AD-A047 394 AIR FORCE WEAPONS LAB KIRTLAND AFB N MEX F/G 22/4
LAUNCH, A COMPUTER CODE FOR DETERMINING LAUNCH VEHICLE RELIABIL--ETC(U)
SEP 77 M A THORNTON

UNCLASSIFIED AFWL-TR-77-126 SBIE-AD-E200 042 NL

1 OF 1
ADA047394

END
DATE
FILMED
1-78
DDC

DDC

AFWL-TR-77-126

AD-E 200 042

AFWL-TR-
77-126

(2)

AD A047394

LAUNCH, A COMPUTER CODE FOR DETERMINING LAUNCH VEHICLE RELIABILITY

September 1977

Final Report

Approved for public release; distribution unlimited

AIR FORCE WEAPONS LABORATORY
Air Force Systems Command
Kirtland Air Force Base, NM 87117

This final report was prepared by the Air Force Weapons Laboratory, Kirtland Air Force Base, New Mexico, Job Order 20070311. Lieutenant Marcia A. Thornton (NSQ) was the Laboratory Project Officer-in-Charge.

This technical report has been reviewed and is approved for publication.

Marcia A. Thornton

MARCIA A. THORNTON
1Lt, USAF
Project Officer

FOR THE COMMANDER

DERMOD KELLEHER
Chief, Power Branch

HERBERT M. FERNANDEZ
Chief, Nuclear Systems Division

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER (14) AFWL-TR-77-126	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) LAUNCH, A COMPUTER CODE FOR DETERMINING LAUNCH VEHICLE RELIABILITY.		5. TYPE OF REPORT & PERIOD COVERED (9) Final Report.
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) MARCIA A./THORNTON 1Lt, USAF Project Officer		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Weapons Laboratory (NSQ) Kirtland Air Force Base, NM 87117		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62601F (16) 20070311 (17) φ3
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Weapons Laboratory (NSQ) Kirtland Air Force Base, NM 87117		12. REPORT DATE (11) Sept[redacted] 1977
		13. NUMBER OF PAGES 76
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) (12) 81p.		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

(18) SBIE

Approved for public release; distribution unlimited

(19) AD-E200 042

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer Code
Launch Vehicle
Missile

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The computer code, LAUNCH, is designed to maintain a data file on the launches of major launch vehicles. From this data file one can obtain a chronological history of any desired vehicle or group of vehicles as well as the historical reliability and/or projected reliability for any future launch. A discussion of the code and its applications are included along with a listing of the program and a sample output results.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

013 150 LB

PREFACE

The work described in this report is part of the total technical support provided by the Power Branch, Nuclear Systems Division of the Air Force Weapons Laboratory (AFWL/NSQ) to the Directorate of Nuclear Surety (AFISC/SN) on the Viking, Lincoln Experimental Satellites 8 and 9, and Mariner Jupiter/Saturn missions.

The author would like to acknowledge the assistance, in the form of a nonlinear regression curve fitting subroutine, provided by the Mathematics Section, Technical Branch, Technology Division of the AFWL.

CONTENTS

ILLUSTRATIONS

SECTION I

INTRODUCTION

The Interagency Nuclear Safety Review Panel (INSRP) is tasked with reviewing the safety analyses of space launches carrying nuclear power sources. After review of the launch safety analysis, the INSRP prepares a Safety Evaluation Report (SER) for submission to the National Security Council and ultimately to the President. The SER contains a recommendation for launch approval or disapproval.

These safety analyses are performed under Energy Research and Development Administration (ERDA) contract and include inputs from many sources including the launch vehicle contractors. For many years the INSRP has accepted the booster vehicle and upper stage failure rates provided by the contractors while wondering how these failure rates compare with the performance history of the vehicle. The Power Branch, Nuclear Systems Division of the Air Force Weapons Laboratory (AFWL) has developed a computer code, LAUNCH, to determine historical failure rates in an effort to resolve the potential differences between contractor supplied and historical failure rates. LAUNCH allows the user to obtain the launch history, historical failure rate, and projected reliability of specific launch vehicles using various reliability growth techniques.

Preliminary results of LAUNCH analysis on Viking, Lincoln Experimental Satellites 8 and 9 (LES 8/9), and Mariner Jupiter/Saturn (MJS) have been incorporated into the SERs for those launches. Similar results for other launches and launch vehicles should prove useful to the INSRP in its review of safety analyses.

SECTION II

COMPUTER CODE

The main program, LAUNCH, is a short bookkeeping program which calls subroutines as directed by data cards input to it. The main program and various subroutines will be discussed individually here. Listings of the main program and subroutines appear in the Appendix. Formats for the data cards are discussed in Section V.

A flowchart of LAUNCH appears as figure 1. The data entry cards are read into the proper arrays. An end-of-file (EOF) card terminates the data entry cards. Subroutine RENMER is called to sort the data entries chronologically and to merge them with the main data file which has been stored on tape or permanent file. The reordered and merged main data file is then written onto tape or permanent file for future use. Data cards indicating the desired output information are read next, and the appropriate subroutines are called. Another EOF card terminates the data cards. This ends the program execution.

RENMER is flowcharted in figure 2. This subroutine uses a "shell" sort to chronologically order the data entries from cards. The reordered set is then written on a scratch file. This scratch file and the main data file are then merged into one single file with any duplicate entries combined into a single entry. Duplicate entries can occur because information is obtained from a variety of sources and more than one source may provide information on a given launch. Program control is returned to LAUNCH to output the requested information using this updated data file.

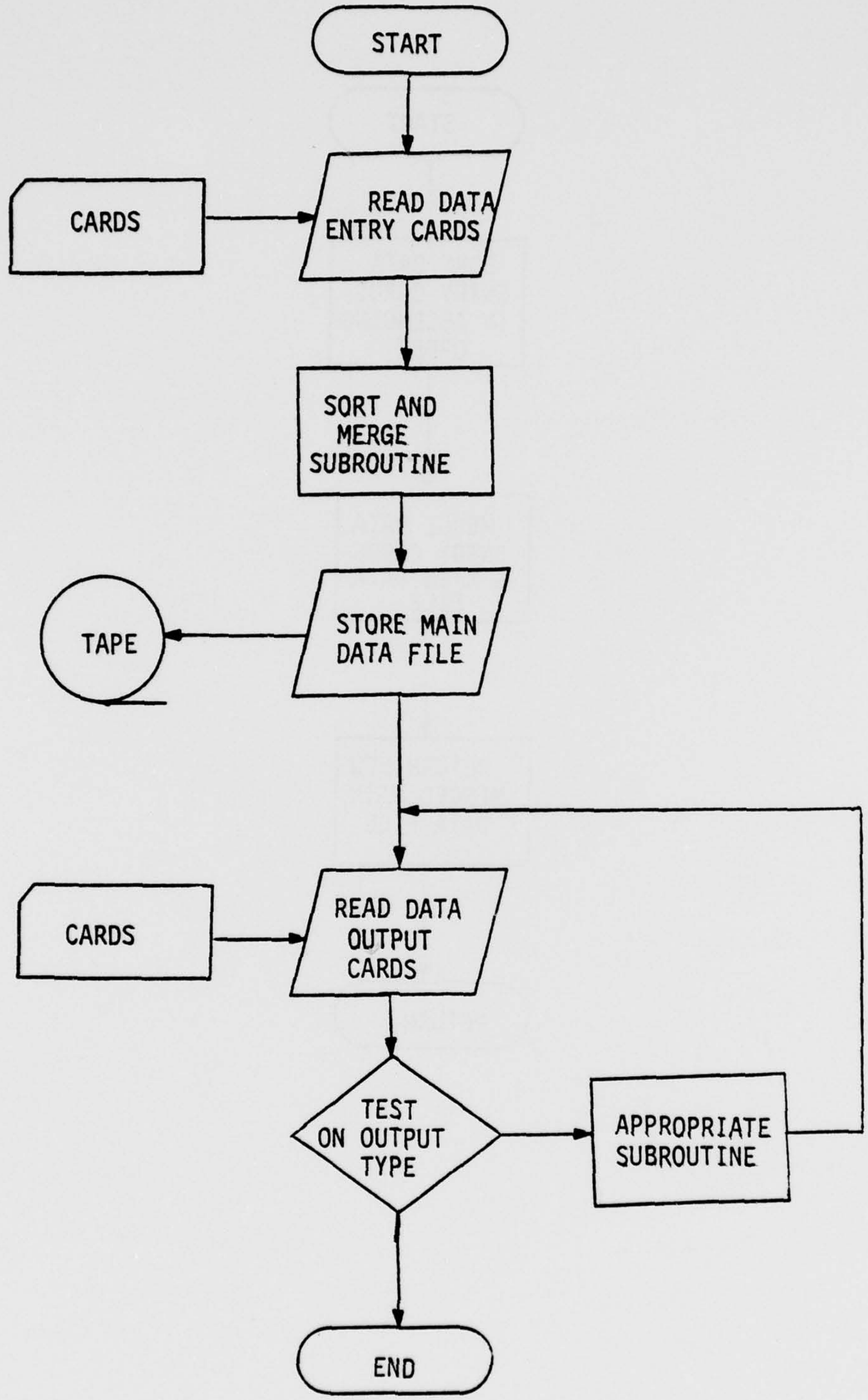

Figure 1. PROGRAM LAUNCH

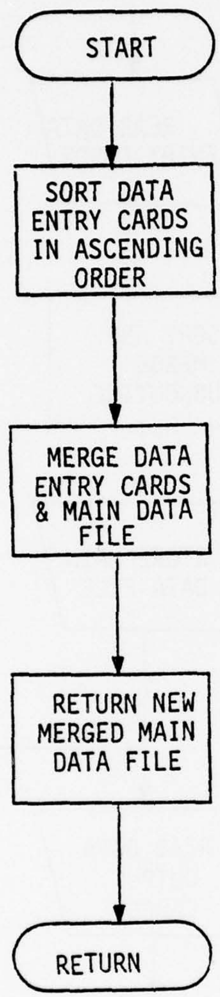

Figure 2. SUBROUTINE RENMER

Subroutine CHANGE appears in figure 3 in flowchart form. CHANGE uses a dummy variable as a temporary storage location for switching the items in the data entries.

A flowchart of VEHICLE appears as figure 4. This subroutine takes the data entries which contain the desired output information from the main data file and enters them into a vehicle array. Data entries are tested to determine if they contain the desired vehicle and, if only failures are desired, if the launch result is a failure. The vehicle array is then printed for use.

Subroutine FAILRAT is flowcharted in figure 5. Using success and failure counters, FAILRAT determines the historical reliability after each launch. This can be done using all launches or for only the last NO launches, where NO is supplied by the user. A success is the successful performance of the desired vehicle; a failure of a booster vehicle is a no-test for the upper stage if the upper stage is the desired vehicle.

Figure 6 is a flowchart of FAILLOC. This subroutine determines the percentage of failures occurring during each launch phase: pad, land, ascent, orbital. These percentages can be for all launches or for only the last NO launches where NO is input by the user.

A flowchart of CURVIT appears as figure 7. CURVIT uses a least-squares nonlinear regression subroutine to determine a best fit to the historical data. Four general equations are currently employed (Y = reliability; X = launch number; A, B, C = curve fit parameters):

$$Y = A + Be^{CX} \qquad (1)$$

$$Y = Ae^{Be^{CX}} \qquad (2)$$

$$Y = A\left(1 - \frac{B}{CX+B}\right) \qquad (3)$$

$$Y = Ae^{B/x} \qquad (4)$$

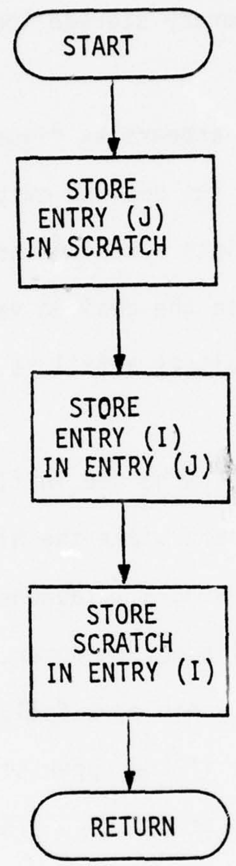

Figure 3. SUBROUTINE CHANGE

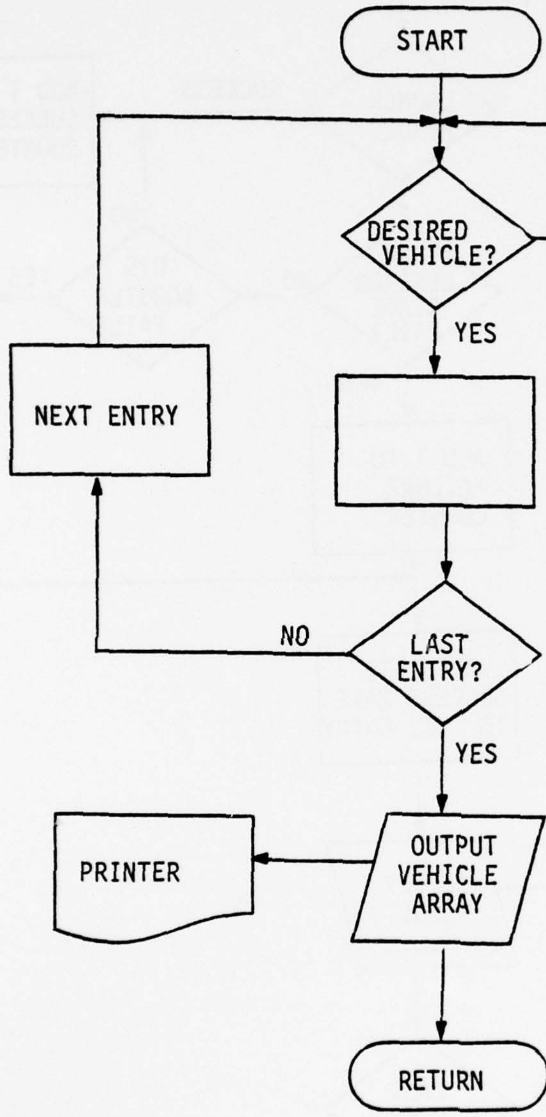

Figure 4. SUBROUTINE VEHICLE

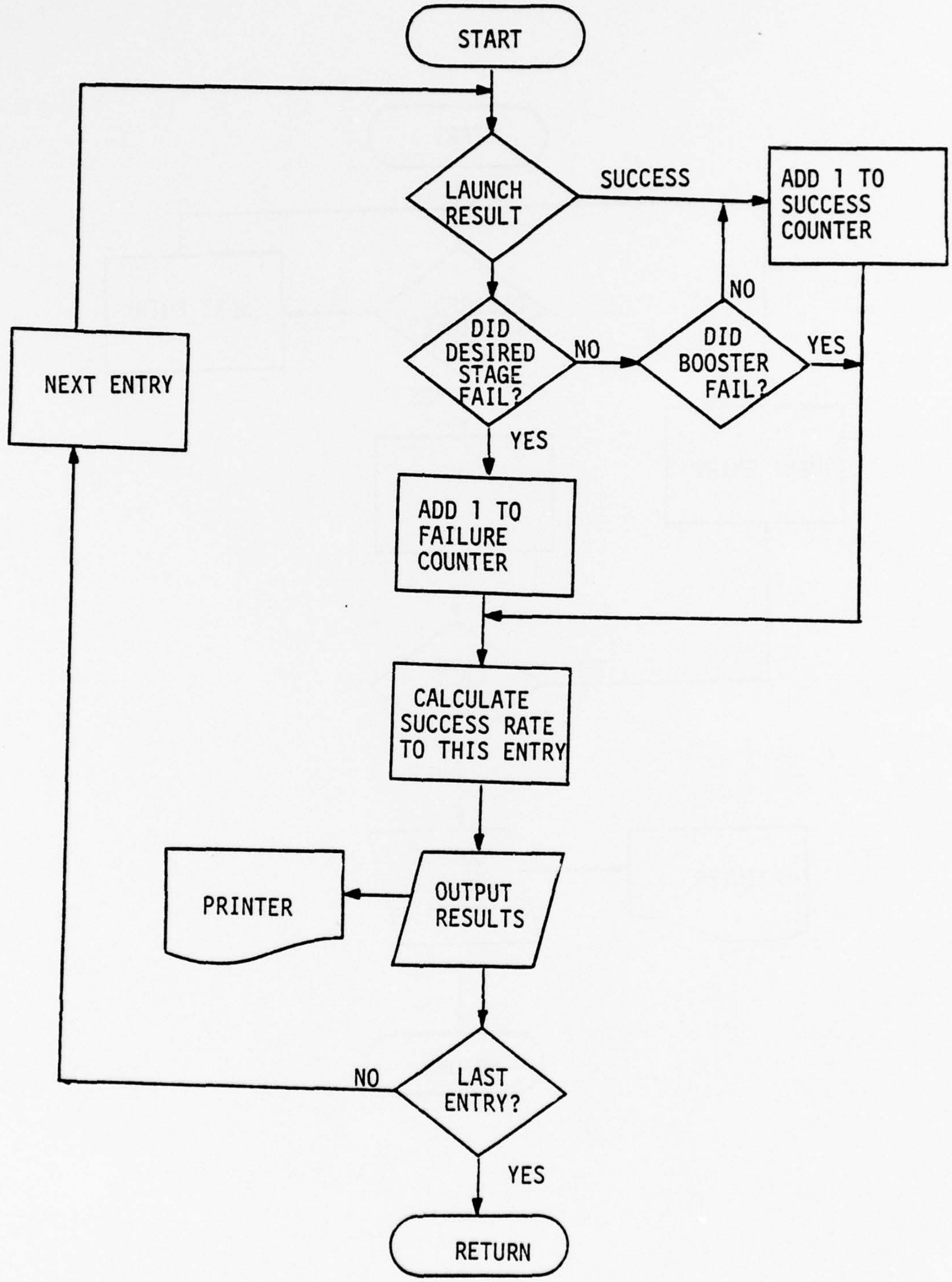

Figure 5. SUBROUTINE FAILRAT

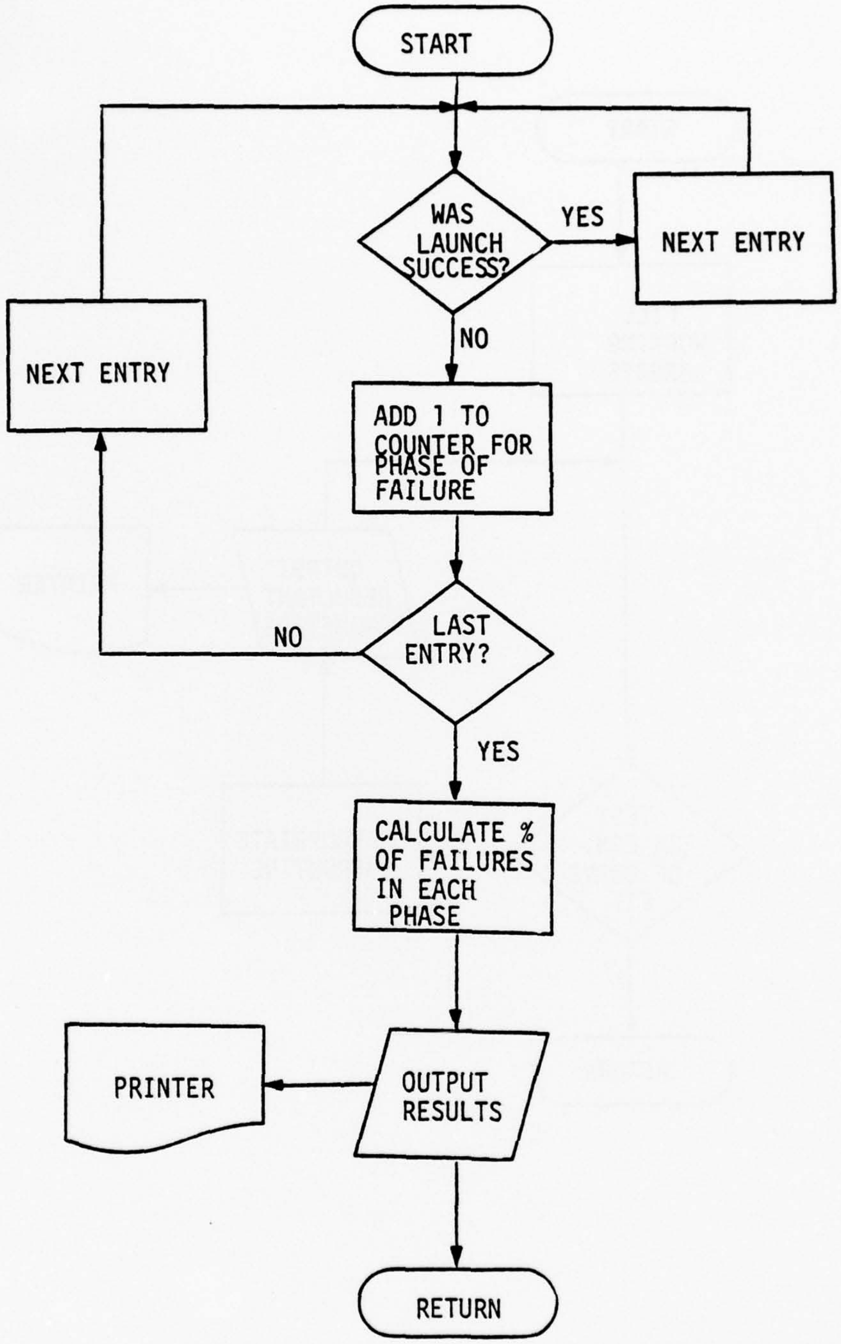

Figure 6. SUBROUTINE FAILLOC

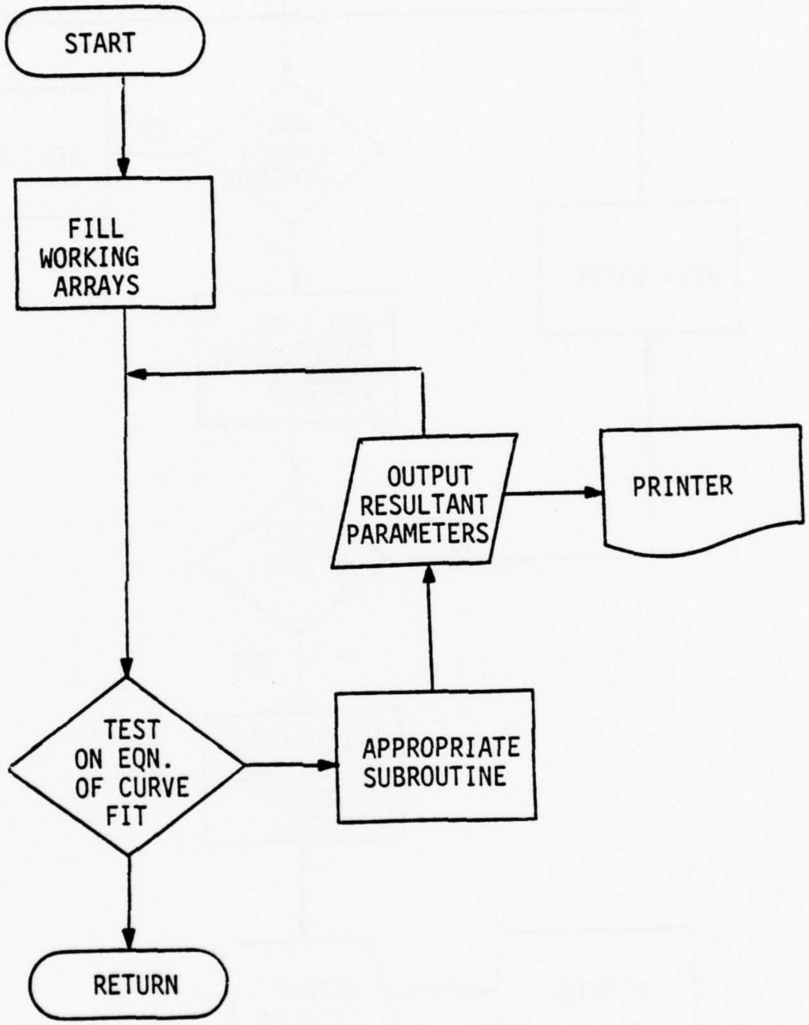

Figure 7. SUBRCUTINE CURVIT

SECTION III

SOURCES OF INFORMATION

The information used to build the data file comes from a variety of sources. The original starting point was the TRW Space Log (reference 2). This provided basic information on launches including date, vehicle, project director, and mission success or failure. Another major source was NASA Pocket Statistics (reference 3) which provided the same type of information on NASA missions. Vandenberg AFB Launch Summary (reference 4) yielded additional data on many boosters which have been used as reentry vehicles rather than space launches. In instances where it was available, contractor information on launch vehicles was also included (reference 5, 6).

SECTION IV

COMPUTER DECK STRUCTURE

The program LAUNCH is written in ANSI standard FORTRAN and requires the appropriate control cards. Different parts of the card deck are separated by end-of-file (EOF) cards (also called end-of-section or end-of-partition). Figure 8 depicts the deck structure.

The control cards must request the storage device for the main data file to read and update the file. They also establish the language used and other peripheral equipment desired.

The main program and associated subroutines follow. Data entry cards are next and may be in any order.

The data output cards must be somewhat organized. First, the vehicle array must be formed (KEY = 0). Then any statistics desired may be determined using that array. If statistics on another vehicle are desired, the vehicle array for that vehicle must be formed first. Any number of vehicles can be examined sequentially during a computer run but only three booster and upper stages can be examined at one time. An EOF card terminates the computer run.

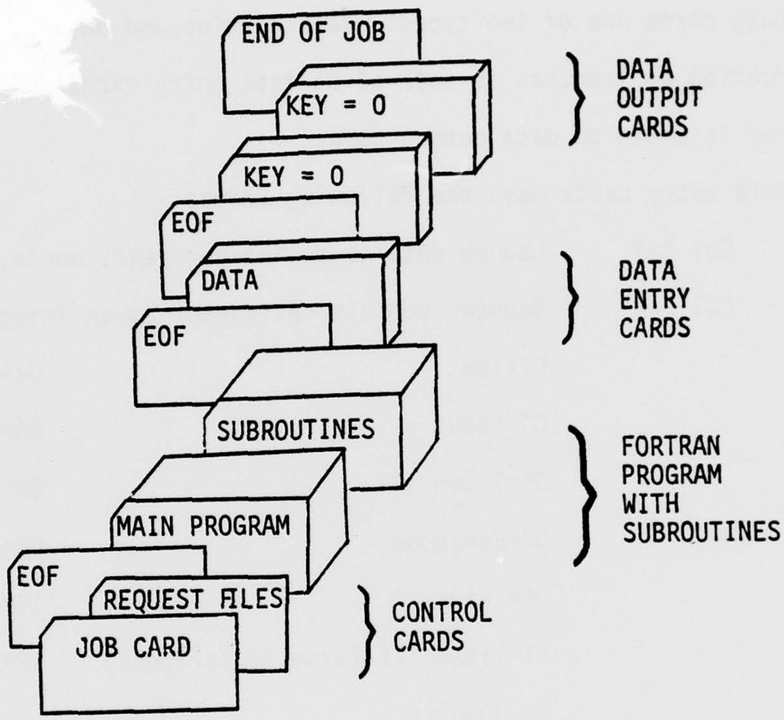

Figure 8. DECK STRUCTURE

SECTION V

DATA CARD FORMAT

Data cards are of two types: data entries and data output. Additional information on launches is entered on data entry cards. The type of output desired is coded on data output cards.

Data entry cards have the following format:

Col 1-6	Launch date as an integer=year, month, day	
Col 7-8	Booster vehicle designator as an integer	
	01=Thor	02=Atlas
	03=Scout	04=Titan II
	05=Titan III	06=Vanguard
	07=Redstone	08=Juno II
	09=Saturn	10=Jupiter
	11=Titan III (w/solid strapons)	12=Titan I
	20=Unknown	
Col 9-10	Upper stage designator as an integer	
	01=Agena	02=Centaur
	03=Able	04=Delta
	05=Burner II	06=Transtage
	10=None	20=Unknown
Col 11-14	Project director as an alphanumeric (e.g., USAF, NASA, USN)	

Col 15 Source of information as an alphanumeric

B=Booster Contractor

E=Air Force Eastern Test Range (AFETR)

N=*NASA Pocket Statistics*

O=Other

T=*TRW Spacelog*

U=Upper Stage Contractor

W=Western Test Range at Vandenberg Air Force Base (VAFB)

Col 16 Launch result as an alphanumeric

F=Failure

S=Success

Col 17 Failure phase as an alphanumeric

A=Ascent

L=Land

O=Orbital

P=Pad

Col 18 Failed stage as an alphanumeric

B=Booster Vehicle

U=Upper Stage

Col 19 Type of launch as an integer

1=Space Launch

2=Training

3=Test

4=Reentry Test Vehicle

5=Suborbital

6=Unknown

Col 20 Multiple launch indicator as an integer

0=First launch of given vehicle on that date

1=One multiple launch of vehicle on that date

The sample data entry card in Figure 9 adds a launch of a Thor Delta spacecraft combination on 13 May 1960 to the main data file. The launch was a NASA space launch which failed during ascent due to an upper state failure. Information is available from Thor contractor.

Data output cards have the following format:

Col 1-2 Output key as an integer

Negative=Automatic call sequence to output complete statistics on desired launch vehicle (this does not form vehicle array)

0=Vehicle history output (this must be done before statistics are requested because the statistics are derived from the vehicle array)

1=Failure rate statistics

2=Failure location statistics

50=Curve fit of data using forms as requested in Cols. 21-24

Col 3-4 Booster vehicle designator (see data entry card Col 7-8) - If zero, any booster may be considered.

6	0	0	5	1	3	0	1	0	4	N	A	S	A	B	F	A	U	1					
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

Figure 9. DATA ENTRY CARD

0	0	0	1	0	1	0	0	0	0		F												
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	15	17	18	19	20	21	22	23	24

Figure 10. DATA OUTPUT CARD

Col 5-6	Upper stage designator (see data entry card Col 9-10) - If zero, any upper stage may be considered.
Col 7-10	Percentage of launches to be used in deriving statistics (zero causes all launches to be considered, negative number causes a summary only to be printed).
Col 11	Blank
Col 12	Launch result to be considered F=Failures only S=All launches
Col 13-14, 17-18	Additional booster vehicle designators
Col 15-16, 19-20	Additional upper stage designators
Col 21-24	Curve-fit forms indicators

Col 21=1	fit to $A+Be^{CX}$
Col 22=1	fit to $Ae^{Be^{CX}}$
Col 23=1	fit to $A\left(1 - \frac{B}{CX+B}\right)$
Col 24=1	fit to $Ae^{B/x}$

The sample data output card in figure 10 requests the vehicle history (failures only) for the Thor Agena spacecraft combination.

SECTION VI

SAMPLE OUTPUT

Figures 11 to 15 show the output generated by LAUNCH. A discussion of each figure will demonstrate its features and uses. The Centaur upper stage will be used as an example.

Figures 11 and 12 are representative of the output from subroutine VEHICLE. The output is labeled for the vehicle being considered and for failures only or all launches. The columns are basically self-explanatory. Abbreviations for the source of information are:

TRW - TRW Space Log

NASA - NASA Pocket Statistics

BV - Booster vehicle contractor

UV - Upper stage contractor

ETR - Air Force Eastern Test Range

WTR - Western Test Range (Vandenberg Air Force Base)

OTHER - Other sources

The launch types are:

SPACE - Orbital flight or space probe

TRNG - Training (these are boosters only in general)

TEST - Vehicle test

RTV - Reentry test vehicle

SUBORB - Suborbital flight

TOTAL FAILURES FOR CENTAUR GDC

DATE	BOOSTER	UPPER STAGE	PROJECT DIRECTOR	SOURCE OF INFORMATION							LAUNCH		FAILURE	
				TRW	NASA	BV	UV	ETR	WTR	OTHER	TYPE	RESULT	STAGE	LOCATION
5/8/62	Atlas	Centaur	NASA	X	X	X	X				Space	F	U	A
6/30/64	Atlas	Centaur	NASA	X	X	X	X				Space	F	U	A
12/11/64	Atlas	Centaur	NASA	X	X	X	X				Space	F	U	O
11/30/70	Atlas	Centaur	NASA	X	X	X	X				Space	F	U	O
5/8/71	Atlas	Centaur	NASA	X	X	X	X				Space	F	U	A
2/11/74	TIII Solid	Centaur	US	X		X	X				Space	F	U	A

Figure 11 Output from Subroutine VEHICLE (Failures Only)

TOTAL VEHICLE HISTORY FOR CENTAUR GDC

DATE	BOOSTER	UPPER STAGE	PROJECT DIRECTOR	SOURCE OF INFORMATION							LAUNCH		FAILURE	
				TRW	NASA	BV	UV	ETR	WTR	OTHER	TYPE	RESULT	STAGE	LOCATION
5/8/62	Atlas	Centaur	NASA	X	X	X	X				Space	F	U	A
11/27/63	Atlas	Centaur	NASA	X	X	X	X				Space	S		
6/30/64	Atlas	Centaur	NASA	X	X	X	X				Space	F	U	A
12/11/64	Atlas	Centaur	NASA	X	X	X	X				Space	F	U	O
3/2/65	Atlas	Centaur	NASA	X	X	X	X				Space	F	B	P
8/11/65	Atlas	Centaur	NASA	X	X	X	X				Space	S		
4/8/66	Atlas	Centaur	NASA		X	X	X				Space	S		
5/30/66	Atlas	Centaur	NASA	X	X	X	X				Space	S		
9/20/66	Atlas	Centaur	NASA	X		X	X				Space	S		
10/26/66	Atlas	Centaur	NASA	X	X	X	X				Space	S		
4/17/67	Atlas	Centaur	NASA	X	X	X	X				Space	S		
7/14/67	Atlas	Centaur	NASA	X	X	X	X				Space	S		
9/8/67	Atlas	Centaur	NASA	X	X	X	X				Space	S		
11/7/67	Atlas	Centaur	NASA	X	X	X	X				Space	S		
1/7/68	Atlas	Centaur	NASA	X	X	X	X				Space	S		
8/10/68	Atlas	Centaur	NASA	X	X	X	X				Space	S		
12/7/68	Atlas	Centaur	NASA	X	X	X	X				Space	S		
2/24/69	Atlas	Centaur	NASA	X	X	X	X				Space	S		
3/27/69	Atlas	Centaur	NASA	X	X	X	X				Space	S		
8/12/69	Atlas	Centaur	NASA	X	X	X	X				Space	S		
11/30/70	Atlas	Centaur	NASA	X	X	X	X				Space	F	U	O
1/25/71	Atlas	Centaur	CSC	X	X	X	X				Space	S		
5/8/71	Atlas	Centaur	NASA	X	X	X	X				Space	F	U	A
5/30/71	Atlas	Centaur	NASA	X	X	X	X				Space	S		
12/19/71	Atlas	Centaur	CSC	X	X	X	X				Space	S		
1/23/72	Atlas	Centaur	CSC	X	X	X	X				Space	S		
3/3/72	Atlas	Centaur	NASA	X	X	X	X				Space	S		
6/13/72	Atlas	Centaur	CSC	X	X	X	X				Space	S		
8/21/72	Atlas	Centaur	NASA	X	X	X	X				Space	S		
4/6/73	Atlas	Centaur	NASA	X	X	X	X				Space	S		
8/23/73	Atlas	Centaur	CSC	X		X	X				Space	S		
11/3/73	Atlas	Centaur	NASA	X		X	X				Space	S		
2/11/74	TIII Solid	Centaur	US	X		X	X				Space	F	U	A
11/21/74	Atlas	Centaur	CSC	X		X	X				Space	S		
12/10/74	TIII Solid	Centaur	Germ	X		X	X				Space	S		
8/20/75	TIII Solid	Centaur	NASA			X	X				Space	S		
9/9/75	TIII Solid	Centaur	NASA	X		X	X				Space	S		
1/15/76	TIII Solid	Centaur	Germ			X	X				Space	S		
1/29/76	Atlas	Centaur	CSC			X	X				Space	S		
5/13/76	Atlas	Centaur	USAF			X	X				Space	S		
7/22/76	Atlas	Centaur	USAF			X	X				Space	S		

Figure 12. Output from Subroutine VEHICLE (All Launches)

Failure locations used are:

P - Failure occurred on launch pad or resulted in a pad impact

L - Failure occurred over land

A - Failure occurred during ascent over water before achieving orbit

O - Failure occurred in orbital phase

Note that the failures in figure 11 are only Centaur failures; the booster failure is not listed. The launches are listed in chronological order and show the basic information at a glance. If more detailed information is desired, it can be obtained from one of the sources marked with an "X".

Figure 13 is a sample output from subroutine FAILRAT. The columns are self-explanatory. Note that the failure of the booster does not affect the success ratio of the upper stage.

The percentages of failures during the various launch phase are shown in figure 14. If the phase of failure is not known, that failure is not considered in determining the percentages.

Figure 15 shows the output of subroutine CURVIT, the reliability growth curve-fitting subprogram. The equations can be plotted as in figure 16 with the historical data to graphically show the trends for the vehicle being considered.

CENTAUR GDC

Adjusted Cumulative Success/Failure Ratio for Last 33 Flights of a Total of 41.

LAUNCH NUMBER	RESULT	STAGE	TOTALS S	TOTALS F	PERCENT
1	F	U	0	1	0.00
2	S		1	1	50.00
3	F	U	1	2	33.33
4	F	U	1	3	25.00
5	F	B	1	3	25.00
6	S		2	3	40.00
7	S		3	3	50.00
8	S		4	3	57.14
9	S		5	3	62.50
10	S		6	3	66.67
11	S		7	3	70.00
12	S		8	3	72.73
13	S		9	3	75.00
14	S		10	3	76.92
15	S		11	3	78.57
16	S		12	3	80.00
17	S		13	3	81.25
18	S		14	3	82.35
19	S		15	3	83.33
20	S		16	3	84.21
21	F	U	16	4	80.00
22	S		17	4	80.95
23	F	U	17	5	77.27
24	S		18	5	78.26
25	S		19	5	79.17
26	S		20	5	80.00
27	S		21	5	80.77
28	S		22	5	81.48
29	S		23	5	82.14
30	S		24	5	82.76
31	S		25	5	83.33
32	S		26	5	83.87
33	F	U	26	6	81.25
34	S		27	5	84.38
35	S		27	5	84.38
36	S		28	4	87.50
37	S		29	3	90.63
38	S		30	3	90.91
39	S		30	3	90.91
40	S		30	3	90.91
41	S		30	3	90.91

Figure 13 Output from Subroutine FAILRAT (Chronological)

CENTAUR GDC

Adjusted Failures Classed by Location for Last 33 Flights of a Total of 41.

LOCATION	FAILURES	PERCENT
Pad	0	0.00
Land	0	0.00
Ascent	2	66.67
Orbital	1	33.33
Total	3	100.00

Figure 14 Output from Subroutine FAILRAT (Summary)

The curve fit for this data is of the form:

Reliability = A * EXP(B * EXP(C * Launch Number))

With the Parameters A, B, C as follows:

A = .8099545744E+00
B = -.8207599285E+00
C = -.9104745067E+00

The curve fit for this data is of the form:

Reliability = A * (1 - B/(C * Launch Number + B))

With the Parameters A, B, C as follows:

A = .8764411049E+00
B = .2880982693E+15
C = .2880982693E+15

The curve fit for this data is of the form:

Reliability = A + B * EXP(C * Launch Number)

With the Parameters A, B, C as follows:

A = .8199671974E+00
B = -.2989172644E+00
C = -.4309589840E+00

The curve fit for this data is of the form:

Reliability = A * EXP(B/Launch Number)

With the Parameters A, B as follows:

A = .8328733128E+00
B = -.3642819210E+00

Figure 15. Output from Subroutine CURVIT

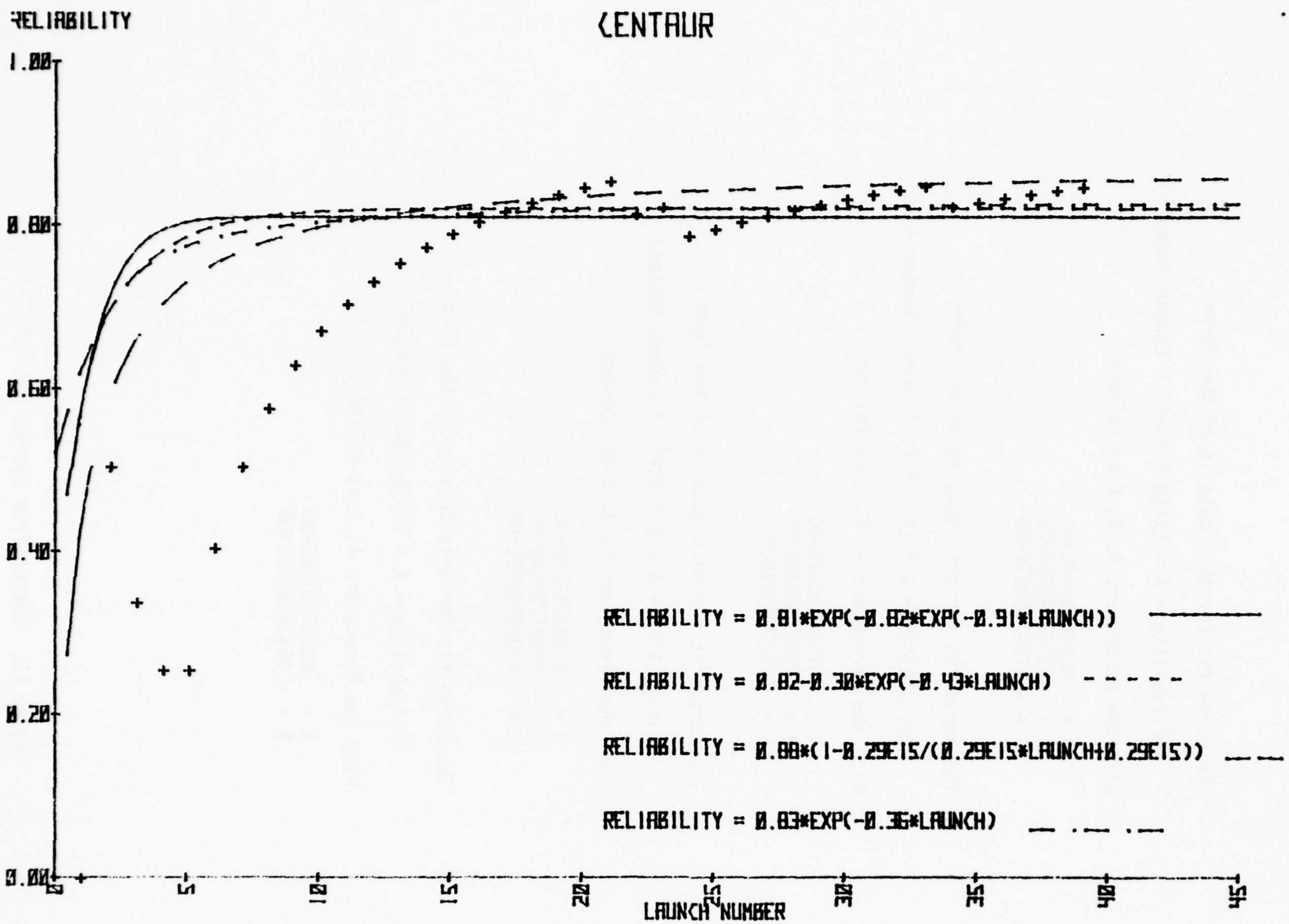

FIGURE 16. PLOT OF EQUATIONS DETERMINED BY SUBROUTINE CURVIT

SECTION VII

RESULTS

LAUNCH has been used to provide preliminary results for use with the safety analyses of the Viking, LES 8/9, and MJS missions. These results give a historical reliability which is used to scale the failure probabilities originated by the analyses.

The booster vehicle for all three missions is the Titan III with strap-on solid rocket motors. This vehicle has had 32 flights and 32 successes. Therefore, it is considered very reliable, and the manufacturer's failure probability of 98% could be used directly.

The Viking and MJS missions use a Centaur upper stage. From Jan 1968 to Jun 1974, there were 21 launches with 19 successes which is a reliability of 90.48%. This is lower than the manufacturer's reliability, and some scaling was done to reconcile the numbers.

LES 8/9 uses a Transtage upper stage. From Jan 1968 to Jun 1974, there were 12 launches with 11 successes which is a reliability of 91.67%. This is also lower than the manufacturer's reliability, and scaling was done before the reliabilities were utilized.

Another consideration is the apportionment of failure probabilities to the launch phases. If all launch vehicles are considered, these figures are:

Pad	5%
Land	5%
Ascent	63%
Orbital	27%

All launches were considered because there have not been sufficient failures of any particular vehicle to accurately predict these percentages.

As information is gathered, more accurate results will be obtained from LAUNCH. These results should prove useful to the INSRP review of nuclear power source launches.

APPENDIX

LISTING OF LAUNCH

The following listing of LAUNCH and its subroutines is provided for the reader who is interested in the details of the program logic. Comments are included to explain the workings of the program and to separate it into logical units.

```
      PROGRAM LAUNCH(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE8,TAPE10,  FITI
     1TAPE3)                                                              FITI
C                                                                         FITI
C        TYPE STATEMENTS FOR THE VARIABLES                                FITI
C                                                                         FITI
      INTEGER ORGDAT                                                      FITI
      INTEGER C1,C2,C3,C4                                                 FITI
      INTEGER BOOST,UPPER,BV,UV,ALN                                       FITI
      INTEGER BV1,BV2,UV1,UV2                                             FITI
      INTEGER SUCC,FAIL                                                   FITI
      REAL LOC                                                            FITI
C                                                                         FITI
C        THE COMMON BLOCKS ARE USED FOR SEPARATE PURPOSES                 FITI
C        BLOCK *DATAIN* CONTAINS THE MAIN DATA FILE                       FITI
C        BLOCK *DATAOUT* CONTAINS THE OUTPUT SPECIFICATIONS               FITI
C        BLOCK *STORIT* CONTAINS HEADING INFORMATION AND VARIABLE COUNTS  FITI
C        BLOCK *FIT* CONTAINS THE CURVE FIT VARIABLES                     FITI
C                                                                         FITI
      COMMON /DATAIN/ ORGDAT,NEWDAT,IDATE(2000),BOOST(2000),UPPER(2000),  FITI
     1SOUR(2000),RES(2000),LOC(2000),PRODIR(2000),ST(2000,7),STG(2000),L  FITI
     2TYP(2000),MULTLA(2000),ALN                                          FITI
C                                                                         FITI
      COMMON /DATAOUT/ BV,UV,BV1,UV1,BV2,UV2,BVEH(20),UVEH(20),BCONT(20)  FITI
     1,UCONT(20),RESU,LATYP(6)                                            FITI
C                                                                         FITI
      COMMON /STORIT/ SUCC(1500),FAIL(1500),RAT(1500),HEAD(12)            FITI
C                                                                         FITI
      COMMON /FIT/ X(2000),Y(2000),WF(2000),WORK(2500),THEOR(2000),NDATA  FITI
     1,RELERR,ABSERR,IFLAG,PAR(6),R,NPAR                                  FITI
C                                                                         FITI
C        THE DATA BLOCKS SET CERTAIN VARIABLES TO ASSIGNED VALUES         FITI
C        LAUNCH TYPE (*LATYP*), BOOSTER VEHICLE (*BVEH*), UPPER STAGE (*  FITI
C        BOOSTER CONTRACTOR (*BCONT*), AND UPPER STAGE CONTRACTOR (*UCON  FITI
C        ASSIGNED                                                         FITI
C                                                                         FITI
      DATA LATYP/6H SPACE,6H  TRNG,6H  TEST,6H   RIV,6HSUBORB,6H      /   FITI
C                                                                         FITI
      DATA BVEH/10HTHOR      ,10HATLAS     ,10HSCOUT     ,10HTITAN II  ,  FITI
     110HTITAN III ,10HVANGUARD  ,10HREDSTONE  ,10HJUNO II   ,10HSATURN   FITI
     2  ,10HJUPITER   ,10HTIII SOLID,10HTITAN I   ,7(10H          ),10H   FITI
     3UNKNOWN   /                                                         FITI
C                                                                         FITI
      DATA UVEH/10HAGENA     ,10HCENTAUR   ,10HABLE      ,10HDELTA     ,  FITI
     110HBURNER II ,10HTRANSTAGE ,13(10H          ),10HUNKNOWN   /        FITI
C                                                                         FITI
      DATA BCONT/10HMDAC      ,10HGDC       ,10HLTV       ,2(10HMMC       FITI
     1 ),5(10H          ),10HMMC       ,9(10H          )/                 FITI
C                                                                         FITI
      DATA UCONT/10H          ,10HGDC       ,10H          ,10HMDAC        FITI
     1,10H          ,10HMMC       ,14(10H          )/                     FITI
C                                                                         FITI
C                                                                         FITI
C        READ THE NEW DATA ENTRY CARDS AND FORM A WORKING ARRAY           FITI
C                                                                         FITI
      I=0                                                                 FITI
    1 I=I+1                                                               FITI
      READ (5,12) IDATE(I),BOOST(I),UPPER(I),PRODIR(I),SOUR(I),RES(I),LO  FITI
```

```
      1G(I),STG(I),LTYP(I),MULTLA(I)                                   FITI
C                                                                      FITI
C         IF THE LAUNCH TYPE IS NOT KNOWN (I.E. EQUALS 0) THE INDICATOR FITI
C         IS SET TO 6 WHICH GIVES A BLANK LAUNCH TYPE IN THE OUTPUT     FITI
C                                                                      FITI
      IF (LTYP(I).EQ.0) LTYP(I)=6                                      FITI
C                                                                      FITI
C         THE DATA ENTRY CARDS ARE TERMINATED BY AN END-OF-FILE CARD    FITI
C                                                                      FITI
      IF (EOF(5)) 4,2                                                  FITI
C                                                                      FITI
C         THE ARRAY *ST* IS BLANKED TO ELIMINATE ERRONEOUS DATA WHEN THE FITI
C         DATA ENTRY CARDS ARE MERGED WITH THE MAIN DATA FILE           FITI
C                                                                      FITI
    2 DO 3 J=1,7                                                       FITI
    3 ST(I,J)=1H                                                       FITI
      GO TO 1                                                          FITI
C                                                                      FITI
    4 NEWDAT=I-1                                                       FITI
C                                                                      FITI
C         SUBROUTINE RENMER IS CALLED TO SORT THE DATA ENTRY CARDS INTO FITI
C         CHRONOLOGICAL ORDER AND MERGE THEM WITH THE MAIN DATA FILE INTO FITI
C         A SINGLE DATA FILE                                            FITI
C                                                                      FITI
      CALL RENMER                                                      FITI
C                                                                      FITI
C         WRITE THE DATA ENTRIES ONTO TAPE OR DISK                      FITI
C                                                                      FITI
      DO 5 I=1,ORGDAT                                                  FITI
      SOUR(I)=1H                                                       FITI
    5 WRITE (8,14) IDATE(I),BOOST(I),UPPER(I),PRODIR(I),(ST(I,J),J=1,7), FITI
     1SOUR(I),RES(I),LOC(I),STG(I),LTYP(I),MULTLA(I)                   FITI
C                                                                      FITI
C         WRITE AN END-OF-FILE MARK AND REWIND THE FILE                 FITI
C                                                                      FITI
      REWIND 8                                                         FITI
C                                                                      FITI
C         DATA OUTPUT CARDS ARE READ FROM THE CARD READER               FITI
C         THESE CARDS INDICATE THE TYPE OF OUTPUT DESIRED               FITI
C                                                                      FITI
    6 READ (5,13) KEY,BV,UV,NO,RESU,BV1,UV1,BV2,UV2,C1,C2,C3,C4        FITI
C                                                                      FITI
C         AN END-OF-FILE CARD TERMINATES THE DATA OUTPUT CARDS          FITI
C                                                                      FITI
      IF (EOF(5)) 11,7                                                 FITI
C                                                                      FITI
C         A TEST ON *KEY* DETERMINES WHICH OUTPUT SUBROUTINES ARE CALLED FITI
C                                                                      FITI
    7 IF (KEY) 10,8,9                                                  FITI
C                                                                      FITI
C         WHEN *KEY*=0 THE VEHICLE ARRAY IS FORMED ACCORDING TO THE     FITI
C         SPECIFICATIONS ON THE DATA OUTPUT CARD                        FITI
C                                                                      FITI
    8 CALL VEHICLE                                                     FITI
      GO TO 6                                                          FITI
C                                                                      FITI
C         WHEN *KEY* IS POSITIVE ITS VALUE DETERMINES WHICH SUBROUTINE IS FITI
```

```
C           CALLED.  THE VALUE OF *NO* IS USED TO DETERMINE THE PERCENTAGE   FITI
C           OF LAUNCHES TO BE CONSIDERED BY THE SUBROUTINES                  FITI
C                                                                            FITI
      9 IF (KEY.EQ.1) CALL FAILRAT (NO)                                      FITI
        IF (KEY.EQ.2) CALL FAILLOC (NO)                                      FITI
        IF (KEY.EQ.50) CALL CURVIT (NO,C1,C2,C3,C4)                          FITI
        GO TO 6                                                              FITI
C                                                                            FITI
C           WHEN *KEY* IS NEGATIVE AN AUTOMATIC SEQUENCE OF SUBROUTINES IS   FITI
C           CALLED.  THE VALUE OF *NO* IS USED TO DETERMINE THE PERCENTAGE   FITI
C           OF LAUNCHES TO BE CONSIDERED BY THE SUBROUTINES.  IN ADDITION    FITI
C           THE SEQUENCE AUTOMATICALLY USES ALL LAUNCHES FOR CALLS TO THE    FITI
C           SUBROUTINES                                                      FITI
C                                                                            FITI
     10 CALL FAILRAT (NO)                                                    FITI
        CALL FAILRAT (J)                                                     FITI
        CALL FAILLOC (NO)                                                    FITI
        CALL FAILLOC (J)                                                     FITI
        CALL CURVIT (NO,1,1,1,1)                                             FITI
        GO TO 6                                                              FITI
C                                                                            FITI
     11 CONTINUE                                                             FITI
C                                                                            FITI
C           THE PROGRAM IS COMPLETED                                         FITI
C                                                                            FITI
        CALL EXIT                                                            FITI
C                                                                            FITI
C           FORMAT STATEMENTS                                                FITI
C                                                                            FITI
C                                                                            FITI
     12 FORMAT (I6,2I2,A4,4A1,2I1)                                           FITI
     13 FORMAT (3I2,I4,A2,4I2,4I1)                                           FITI
     14 FORMAT (I6,2I2,A4,11A1,2I1)                                          FITI
        END                                                                  FITI
```

```
          SUBROUTINE RENMER                                               FI
C                                                                         FI
C           TYPE STATEMENTS                                               FI
C                                                                         FI
          REAL LOC                                                        FI
          INTEGER ORGDAT                                                  FI
          INTEGER BOOST,UPPER                                             FI
          DIMENSION V15(7), V55(7)                                        FI
C                                                                         FI
C          COMMON BLOCKS                                                  FI
C                                                                         FI
          COMMON /DATAIN/ ORGDAT,NEWDAT,IDATE(2000),BOOST(2000),UPPER(2000), FI
         1SOUR(2000),RES(2000),LOC(2000),PRODIR(2000),ST(2000,7),STG(2000),L FI
         2TYP(2000),MULTLA(2000),ALN                                      FI
C                                                                         FI
C           PERFORM A *SHELL* SORT ON THE DATA ENTRY CARDS TO ORDER THEM  FI
C           CHRONOLOGICALLY                                               FI
C                                                                         FI
          N=NEWDAT                                                        FI
        1 N=N/2                                                           FI
          IF (N.EQ.0) GO TO 10                                            FI
          K=NEWDAT-N                                                      FI
          J=1                                                             FI
        2 I=J                                                             FI
        3 IF (IDATE(I)-IDATE(I+N)) 9,4,8                                  FI
        4 IF (BOOST(I)-BOOST(I+N)) 9,5,8                                  FI
        5 IF (UPPER(I)-UPPER(I+N)) 9,6,8                                  FI
        6 IF (PRODIR(I)-PRODIR(I+N)) 9,7,8                                FI
        7 IF (MULTLA(I)-MULTLA(I+N)) 9,9,8                                FI
C                                                                         FI
C           SUBROUTINE *CHANGE* CHANGES ENTRY *I* WITH ENTRY *I+N*        FI
C                                                                         FI
        8 CALL CHANGE (I,I+N)                                             FI
          I=I-N                                                           FI
          IF (I.GE.1) GO TO 3                                             FI
        9 J=J+1                                                           FI
          IF (J-K) 2,2,1                                                  FI
C                                                                         FI
C           WRITE THE REORDERED ENTRIES ON A FILE FOR COMPARISON WITH THE FI
C           MAIN DATA FILE                                                FI
C                                                                         FI
       10 DO 11 I=1,NEWDAT                                                FI
       11 WRITE (3,34) IDATE(I),BOOST(I),UPPER(I),PRODIR(I),(ST(I,J),J=1,7), FI
         1SOUR(I),RES(I),LOC(I),STG(I),LTYP(I),MULTLA(I)                  FI
          REWIND 3                                                        FI
C                                                                         FI
C           MERGE THE MAIN DATA FILE AND THE REORDERED DATA ENTRIES INTO A FI
C           SINGLE FILE WITH NO DUPLICATE ENTRIES (POSSIBLE SINCE MANY    FI
C           SOURCES OF INFORMATION ARE USED)                              FI
C           THE MAIN DATA FILE IS ON TAPE10                               FI
C           THE DATA ENTRIES ARE ON TAPE3                                 FI
C                                                                         FI
          IN=3                                                            FI
          IO=10                                                           FI
C                                                                         FI
C           SET COUNTER TO ONE                                            FI
C                                                                         FI
```

```
      I=1
C
C     SET SAME FILE INDICATOR TO ZERO
C
      IFLAG=0
C
C     IF NO DATA ENTRY CARDS HAVE BEEN READ, COPY DATA FILE FROM
C     TAPE10 TO TAPE8
C     SET SAME FILE INDICATOR TO ONE
C
      IF (NEWDAT.EQ.0) IFLAG=1
      IF (NEWDAT.EQ.0) IN=10
C
C     READ FILE
C
      READ (IN,34) IV51,IV52,IV53,V54,(V55(II),II=1,7),V56,V57,V58,V59,I
     1V60,IV61
C
C     CHECK FOR END-OF-FILE
C
      IF (EOF(IN)) 32,12
C
C     READ FILE
C
   12 READ (IO,34) IV11,IV12,IV13,V14,(V15(II),II=1,7),V16,V17,V18,V19,I
     1V20,IV21
C
C     CHECK FOR END-OF-FILE
C
      IF (EOF(IO)) 31,13
C
C     COMPARE LAUNCH DATES -
C        IF EQUAL, COMPARE FURTHER
C        IF NOT EQUAL, FILL DATA ARRAY WITH EARLIER LAUNCH ENTRY
C
   13 IF (IV11-IV51) 24,16,14
C
C     MAIN DATA FILE HAS EARLIER LAUNCH
C
   14 IDATE(I)=IV51
      BOOST(I)=IV52
      UPPER(I)=IV53
      PRODIR(I)=V54
      DO 15 J=1,7
   15 SI(I,J)=V55(J)
      IF (V56.EQ.1HT) SI(I,1)=1HX
      IF (V56.EQ.1HN) SI(I,2)=1HX
      IF (V56.EQ.1HB) SI(I,3)=1HX
      IF (V56.EQ.1HU) SI(I,4)=1HX
      IF (V56.EQ.1HE) SI(I,5)=1HX
      IF (V56.EQ.1HW) SI(I,6)=1HX
      IF (V56.EQ.1HO) SI(I,7)=1HX
      RES(I)=1HS
      IF (V57.EQ.1HF) RES(I)=V57
      LOC(I)=V58
      STG(I)=V59
      LIYR(I)=IV60
```

```
      MULTLA(I)=IV61
C
C        INCREASE COUNTER
C
      I=I+1
C
C        READ FILE
C
      READ (IN,34) IV51,IV52,IV53,V54,(V55(II),II=1,7),V56,V57,V58,V59,I
     1V60,IV61
C
C        CHECK FOR END-OF-FILE
C
      IF (EOF(IN)) 32,13
C
C        DATA ENTRY HAS EARLIER LAUNCH
C
   16 IDATE(I)=IV11
      BOOST(I)=IV12
      UPPER(I)=IV13
      PRODIR(I)=V14
      DO 17 J=1,7
   17 ST(I,J)=V15(J)
      IF (V16.EQ.1HT) ST(I,1)=1HX
      IF (V16.EQ.1HN) ST(I,2)=1HX
      IF (V16.EQ.1HB) ST(I,3)=1HX
      IF (V16.EQ.1HU) ST(I,4)=1HX
      IF (V16.EQ.1HE) ST(I,5)=1HX
      IF (V16.EQ.1HW) ST(I,6)=1HX
      IF (V16.EQ.1HO) ST(I,7)=1HX
      RES(I)=1HS
      IF (V17.EQ.1HF) RES(I)=V17
      LOC(I)=V18
      STG(I)=V19
      LTYP(I)=IV20
      MULTLA(I)=IV21
C
C        INCREASE COUNTER
C
      I=I+1
C
C        READ FILE
C
      READ (IO,34) IV11,IV12,IV13,V14,(V15(II),II=1,7),V16,V17,V18,V19,I
     1V20,IV21
C
C        CHECK FOR END-OF-FILE
C
      IF (EOF(IO)) 31,13
C
C        CHECK FOR THE SAME BOOSTER VEHICLE
C
   18 IF (IV12-IV52) 16,19,14
C
C        CHECK FOR THE SAME UPPER STAGE
C
   19 IF (IV13-IV53) 16,20,14
```

```
      C
      C        CHECK FOR THE SAME PROJECT DIRECTOR
      C
       20 IF (V14-V54) 16,21,14
      C
      C        CHECK FOR MULTIPLE LAUNCH OF SAME VEHICLE
      C
       21 IF (IV21-IV61) 16,22,14
      C
      C        COMBINE THE DUPLICATE INFORMATION INTO A SINGLE ENTRY
      C
       22 IDATE(I)=IV11
          BOOST(I)=IV12
          UPPER(I)=IV13
          PRODIR(I)=V14
          DO 23 J=1,7
          ST(I,J)=V15(J)
          IF (V55(J).NE.1H ) ST(I,J)=V55(J)
       23 CONTINUE
          IF (V16.EQ.1HT.OR.V56.EQ.1HT) ST(I,1)=1HX
          IF (V16.EQ.1HN.OR.V56.EQ.1HN) ST(I,2)=1HX
          IF (V16.EQ.1HB.OR.V56.EQ.1HB) ST(I,3)=1HX
          IF (V16.EQ.1HU.OR.V56.EQ.1HU) ST(I,4)=1HX
          IF (V16.EQ.1HE.OR.V56.EQ.1HE) ST(I,5)=1HX
          IF (V16.EQ.1HW.OR.V56.EQ.1HW) ST(I,6)=1HX
          IF (V16.EQ.1HO.OR.V56.EQ.1HO) ST(I,7)=1HX
          RES(I)=1HS
          IF (V17.EQ.1HF.OR.V57.EQ.1HF) RES(I)=1HF
          LOC(I)=1H
          IF (V18.NE.1H ) LOC(I)=V18
          IF (V58.NE.1H ) LOC(I)=V58
          STG(I)=1H
          IF (V19.NE.1H ) STG(I)=V19
          IF (V59.NE.1H ) STG(I)=V59
          LTYP(I)=6
          IF (IV20.NE.6) LTYP(I)=IV20
          IF (IV60.NE.6) LTYP(I)=IV60
          MULTLA(I)=0
          IF (IV21.NE.0) MULTLA(I)=IV21
          IF (IV61.NE.0) MULTLA(I)=IV61
      C
      C        INCREASE COUNTER
      C
          I=I+1
      C
      C        READ FILE
      C
          READ (IN,34) IV51,IV52,IV53,V54,(V55(II),II=1,7),V56,V57,V58,V59,I
         1V60,IV61
      C
      C        CHECK FOR END-OF-FILE
      C
          IF (EOF(IN)) 32,24
      C
      C        CHECK FOR DUPLICATE ENTRY
      C
       24 IF (IDATE(I-1)-IV51) 34,25,30
```

```
      25 IF (BOOST(I-1)-IV52) 30,26,30                                F
      26 IF (UPPER(I-1)-IV53) 30,27,30                                F
      27 IF (PRODIR(I-1)-V54) 30,28,30                                F
      28 IF (MULILA(I-1)-IV61) 30,29,30                               F
      29 I=I-1                                                        F
         GO TO 22                                                     F
C                                                                     F
C        READ FILE                                                    F
C                                                                     F
      30 READ (IO,34) IV11,IV12,IV13,V14,(V15(II),II=1,7),V16,V17,V18,V19,I F
        1V20,IV21                                                     F
C                                                                     F
C        CHECK FOR END-OF-FILE                                        F
C                                                                     F
         IF (EOF(IO)) 31,13                                           F
C                                                                     F
C        CHECK DUPLICATE FILE INDICATOR TO SEE IF BOTH END-OF-FILE MARKS F
C        HAVE BEEN READ                                               F
C                                                                     F
      31 IF (IFLAG.NE.0) GO TO 33                                     F
C                                                                     F
C        CHANGE NUMBER OF FILE BEING READ SO THAT ONLY ONE FILE IS BEING F
C        USED                                                         F
C                                                                     F
         IO=3                                                         F
C                                                                     F
C        SET DUPLICATE INDICATOR TO ONE TO INDICATE THAT ONE END-OF-FILE F
C        HAS BEEN READ                                                F
C                                                                     F
         IFLAG=1                                                      F
C                                                                     F
C        READ FILE                                                    F
C                                                                     F
         READ (IO,34) IV11,IV12,IV13,V14,(V15(II),II=1,7),V16,V17,V18,V19,I F
        1V20,IV21                                                     F
C                                                                     F
C        CHECK FOR END-OF-FILE                                        F
C                                                                     F
         IF (EOF(IO)) 33,13                                           F
C                                                                     F
C        CHECK DUPLICATE FILE INDICATOR TO SEE IF BOTH END-OF-FILE MARKS F
C        HAVE BEEN READ                                               F
C                                                                     F
      32 IF (IFLAG.NE.0) GO TO 33                                     F
C                                                                     F
C        CHANGE NUMBER OF FILE BEING READ SO THAT ONLY ONE FILE IS BEING F
C        USED                                                         F
C                                                                     F
         IN=10                                                        F
C                                                                     F
C        SET DUPLICATE INDICATOR TO ONE TO INDICATE THAT ONE END-OF-FILE F
C        HAS BEEN READ                                                F
C                                                                     F
         IFLAG=1                                                      F
C                                                                     F
C        READ FILE                                                    F
C                                                                     F
```

```
      READ (IN,34) IV51,IV52,IV53,V54,(V55(II),II=1,7),V56,V57,V58,V59,I   F1
     1V60,IV61                                                             F1
C                                                                          F1
C        CHECK FOR END-OF-FILE                                             F1
C                                                                          F1
      IF (EOF(IN)) 33,13                                                   F1
C                                                                          F1
C        SET *ORGDAT* EQUAL TO THE NUMBER OF ACTUAL DATA ENTRIES IN THE    F1
C        MAIN DATA FILE                                                    F1
C                                                                          F1
   33 ORGDAT=I-1                                                           F1
C                                                                          F1
C        RETURN CONTROL TO MAIN PROGRAM                                    F1
C                                                                          F1
      RETURN                                                               F1
C                                                                          F1
C        FORMAT STATEMENTS                                                 F1
C                                                                          F1
C                                                                          F1
   34 FORMAT (I6,2I2,A4,11A1,2I1)                                          F1
      END                                                                  F1
```

```
      SUBROUTINE CHANGE (I,J)
C
C        TYPE STATEMENTS
C
      REAL LOG
      INTEGER ORGDAT
      INTEGER BOOST,UPPER
C
C        COMMON BLOCKS
C
      COMMON /DATAIN/ ORGDAT,NEWDAT,IDATE(2000),BOOST(2000),UPPER(2000),
     1SOUR(2000),RES(2000),LOC(2000),PRODIR(2000),ST(2000,7),STG(2000),L
     2TYP(2000),MULTLA(2000),ALN
C
C        THIS SUBROUTINE USES A DUMMY VARIABLE *SCR* TO SWITCH ENTRY *I*
C        ENTRY *J*
C
      SCR=IDATE(J)
      IDATE(J)=IDATE(I)
      IDATE(I)=SCR
C
      SCR=BOOST(J)
      BOOST(J)=BOOST(I)
      BOOST(I)=SCR
C
      SCR=UPPER(J)
      UPPER(J)=UPPER(I)
      UPPER(I)=SCR
C
      SCR=SOUR(J)
      SOUR(J)=SOUR(I)
      SOUR(I)=SCR
C
      SCR=RES(J)
      RES(J)=RES(I)
      RES(I)=SCR
C
      SCR=LOC(J)
      LOC(J)=LOC(I)
      LOC(I)=SCR
C
      SCR=PRODIR(J)
      PRODIR(J)=PRODIR(I)
      PRODIR(I)=SCR
C
      DO 1 K=1,7
      SCR=ST(J,K)
      ST(J,K)=ST(I,K)
    1 ST(I,K)=SCR
C
      SCR=STG(J)
      STG(J)=STG(I)
      STG(I)=SCR
C
      SCR=LTYP(J)
      LTYP(J)=LTYP(I)
      LTYP(I)=SCR
```

SUBROUTINE CHANGE 74/74 OPT=1 FTN 4.6+433 08/2

```
      C                                                                  FI
            SCR=MULTLA(J)                                                FI
            MULTLA(J)=MULTLA(I)                                          FI
            MULTLA(I)=SCR                                                FI
      C                                                                  FI
            RETURN                                                       FI
            END                                                          FI
```

```
      SUBROUTINE VEHICLE
C
C        TYPE STATEMENTS
C
      INTEGER SUCC,FAIL
      INTEGER ORGDAT
      INTEGER BOOST,UPPER,BV,UV,BV1,UV1,BV2,UV2,ALN
      REAL LOG
      DIMENSION V5(7)
C
C        COMMON BLOCKS
C
      COMMON /DATAIN/ ORGDAT,NEWDAT,IDATE(2000),BOOST(2000),UPPER(2000),
     1SOUR(2000),RES(2000),LOG(2000),PRODIR(2000),ST(2000,7),STG(2000),L
     2TYP(2000),MULTLA(2000),ALN
C
      COMMON /DATAOUT/ BV,UV,BV1,UV1,BV2,UV2,BVEH(20),UVEH(20),BCONT(20)
     1,UCONT(20),RESU,LATYP(6)
C
      COMMON /STORIT/ SUCC(1500),FAIL(1500),RAT(1500),HEAD(12)
C
C        BLANK THE HEADING ARRAY
C
      DO 1 I=1,12
    1 HEAD(I)=10H
C
C        FILL THE HEADING ARRAY
C
      IF (BV.EQ.0.AND.UV.EQ.0) HEAD(1)=10HALL LAUNCH
      IF (BV.EQ.0.AND.UV.EQ.0) HEAD(2)=10HVEHICLES
      IF (BV.NE.0) HEAD(1)=BVEH(BV)
      IF (UV.NE.0) HEAD(2)=UVEH(UV)
      IF (BV.NE.0) HEAD(3)=BCONT(BV)
      IF (UV.NE.0) HEAD(4)=UCONT(UV)
      IF (BV1.NE.0) HEAD(5)=BVEH(BV1)
      IF (UV1.NE.0) HEAD(6)=UVEH(UV1)
      IF (BV1.NE.0) HEAD(7)=BCONT(BV1)
      IF (UV1.NE.0) HEAD(8)=UCONT(UV1)
      IF (BV2.NE.0) HEAD(9)=BVEH(BV2)
      IF (UV2.NE.0) HEAD(10)=UVEH(UV2)
      IF (BV2.NE.0) HEAD(11)=BCONT(BV2)
      IF (UV2.NE.0) HEAD(12)=UCONT(UV2)
C
C        ZERO INDEX FOR VEHICLE ARRAY
C
      J=0
C
C        TEST EACH ENTRY IN MAIN DATA FILE TO DETERMINE IF IT MEETS
C        OUTPUT SPECIFICATIONS ON VEHICLE TYPE AND LAUNCH RESULT
C
      DO 6 I=1,ORGDAT
      READ (8,17) IV1,IV2,IV3,V4,(V5(J),J=1,7),V6,V7,V8,V9,IV10,IV11
C
C        TEST FOR END-OF-FILE
C
      IF (EOF(8)) 6,2
C
```

```
C        TEST FOR VEHICLE TYPE TO BE OUTPUT
C              IF *BV* AND *UV* ARE BOTH ZERO, OUTPUT ALL VEHICLES
C              IF *BV* OR *UV* NOT EQUAL ZERO, THEN OUTPUT ONLY DESIRED
C              VEHICLES
C
    2 IF (BV.EQ.0.AND.UV.EQ.0) GO TO 4
      IF (IV2.EQ.BV.OR.IV2.EQ.BV1.OR.IV2.EQ.BV2) GO TO 3
      IF ((IV3.EQ.UV.OR.IV3.EQ.UV1.OR.IV3.EQ.UV2).AND.BV.EQ.0) GO TO 4
      GO TO 6
    3 IF (UV.EQ.0) GO TO 4
      IF (IV3.EQ.UV.OR.IV3.EQ.UV1.OR.IV3.EQ.UV2) GO TO 4
      GO TO 6
C
C        CHECK TO SEE IF ONLY FAILURES ARE DESIRED
C
    4 IF (RESU.EQ.2H F.AND.V7.NE.1HF) GO TO 6
      IF (RESU.EQ.2H F.AND.BV.EQ.0.AND.UV.NE.0.AND.V9.NE.1HU) GO TO 6
      IF (RESU.EQ.2H F.AND.BV.NE.0.AND.UV.EQ.0.AND.V9.NE.1HB) GO TO 6
C
C        INCREMENT VEHICLE ARRAY INDEX AND FILL VEHICLE ARRAY
C
      J=J+1
      IDATE(J)=IV1
      BOOST(J)=IV2
      UPPER(J)=IV3
      PROBIR(J)=V4
      DO 5 K=1,7
    5 ST(J,K)=V5(K)
      RES(J)=V7
      LOC(J)=V8
      STG(J)=V9
      LTYP(J)=IV10
      ALN=J
    6 CONTINUE
C
C        REWIND THE MAIN DATA FILE
C
      REWIND 8
C
C        IF NO FAILURES HAVE BEEN FOUND TO FILL THE ARRAY A MESSAGE IS
C        PRINTED AND CONTROL IS RETURNED TO THE MAIN PROGRAM
C
      IF (J.GT.0) GO TO 7
      PRINT 16
      RETURN
C
C        THE VEHICLE ARRAY IS PRINTED AFTER SEPARATING THE DATE INTO
C        MONTH/DAY/YEAR
C
    7 DO 9 I=1,ALN
      I03=IDATE(I)/10000
      I01=(IDATE(I)-I03*10000)/100
      I02=IDATE(I)-I01*100-I03*10000
      IF (MOD((I-1),50).NE.0) GO TO 8
      WRITE (6,10)
      IF (RESU.EQ.2H S) WRITE (6,11)
      IF (RESU.EQ.2H F) WRITE (6,12)
```

```
      WRITE (6,13) HEAD                                                    F
      WRITE (6,14)                                                         F
    8 WRITE (6,15) ID1,ID2,ID3,BVEH(BOOST(I)),UVEH(UPPER(I)),PRODIR(I),(   F
     1ST(I,J),J=1,7),LATYP(LTYP(I)),RES(I),STG(I),LOC(I)                   F
    9 CONTINUE                                                             F
C                                                                          F
C     RETURN CONTROL TO THE MAIN PROGRAM                                   F
C                                                                          F
      RETURN                                                               F
C                                                                          F
C     FORMAT STATEMENTS                                                    F
C                                                                          F
C                                                                          F
   10 FORMAT (1H1)                                                         F
   11 FORMAT (T40,25HTOTAL VEHICLE HISTORY FOR)                            F
   12 FORMAT (T40,18HTOTAL FAILURES FOR)                                   F
   13 FORMAT (3(T40,A10,2X,A10/T42,A10,2X,A10/))                           F
   14 FORMAT (T4,4HDATE,T12,7HBOOSTER,T24,11HUPPER STAGE,T39,16HPROJECT    F
     1DIRECTOR,T55,21HSOURCE OF INFORMATION,T99,6HLAUNCH,T114,7HFAILURE/   F
     2T54,35HTRW  NASA  DV  UV  ETR  WTR OTHER,T96,13HTYPE    RESULT,T1    F
     311,15HSTAGE  LOCATION/)                                              F
   15 FORMAT (1X,I2,1H/,I2,1H/,I2,T12,A10,T24,A10,T43,A4,T59,A1,T64,A1,T   F
     169,A1,T74,A1,T80,A1,T85,A1,T90,A1,T95,A6,T105,A1,T113,A1,T121,A1)    F
   16 FORMAT (1H1,T45,30HNO FAILURES HAVE BEEN REPORTED)                   F
   17 FORMAT (I6,2I2,A4,11A1,2I1)                                          F
      END                                                                  F
```

```
      SUBROUTINE FAILRAT (NO)
C
C         TYPE STATEMENTS
C
      INTEGER ALN,BV,UV
      INTEGER SUCC,FAIL
      INTEGER BV1,BV2,UV1,UV2
C
C        COMMON BLOCKS
C
      COMMON /DATAIN/ ORGDAT,NEWDAT,IDATE(2000),BOOST(2000),UPPER(2000),
     1SOUR(2000),RES(2000),LOC(2000),PRODIN(2000),ST(2000,7),STG(2000),L
     2TYP(2000),MULTLA(2000),ALN
C
      COMMON /DATAOUT/ BV,UV,BV1,UV1,BV2,UV2,BVEH(20),UVEH(20),BCONT(20)
     1,UCONT(20),RESU,LATYP(5)
C
      COMMON /STORIT/ SUCC(1500),FAIL(1500),RAT(1500),HEAD(12)
C
C         ZERO THE SUCCESS AND FAILURE COUNTERS
C
      DO 1 I=1,ALN
      SUCC(I)=0
    1 FAIL(I)=0
C
C         DETERMINE THE NUMBER OF LAUNCHES TO BE USED IN CALCULATIONS
C
      NO1=NO*ALN/100
      IF (NO.EQ.0) NO1=ALN
      IF (NO.LT.0) NO1=-NO1
C
C         THE NUMBER OF SUCCESSES AND FAILURES ARE CALCULATED
C         THESE MAY BE THE TOTALS TO THE LAUNCH BEING CONSIDERED OR ONLY
C         THE LAST *NO* LAUNCHES
C         A FAILURE OF THE BOOSTER WHEN THE UPPER STAGE IS THE DESIRED
C         VEHICLE RESULTS IN A NO-TEST CONDITION FOR THE UPPER STAGE
C
      DO 4 I=1,ALN
      DO 3 J=1,NO1
      IF (I+J.GT.ALN+1) GO TO 3
      IF (STG(I).EQ.1HB.AND.BV.NE.0) GO TO 2
      IF (STG(I).EQ.1HU.AND.UV.NE.0) GO TO 2
      IF (STG(I).EQ.1HB.AND.BV.EQ.0.AND.UV.NE.0) GO TO 3
C
C         INCREMENT THE SUCCESS COUNTER
C
      SUCC(I+J-1)=SUCC(I+J-1)+1
      GO TO 3
C
C         INCREMENT THE FAILURE COUNTER
C
    2 FAIL(I+J-1)=FAIL(I+J-1)+1
    3 CONTINUE
C
C         AFTER EACH LAUNCH THE SUCCESS RATIO IS CALCULATED
C
      RAT(I)=SUCC(I)*100.0/(SUCC(I)+FAIL(I))
```

```
      IF (NG.LT.0) GO TO 4                                              FI
C                                                                       FI
C        IF A RUNNING TALLY IS REQUESTED IT IS PRINTED AT THIS POINT    FI
C                                                                       FI
      IF (MOD((I-1),50).EQ.0) WRITE (6,5) HEAD,NO1,ALN                  FI
      WRITE (6,6) I,RES(I),SIG(I),SUCC(I),FAIL(I),RAT(I)                FI
    4 CONTINUE                                                          FI
      IF (NG.GE.0) RETURN                                               FI
C                                                                       FI
C        IF ONLY A SUMMARY IS REQUESTED IT PRINTED AT THIS POINT        FI
C                                                                       FI
      WRITE (6,7) NO1,ALN,SUCC(ALN),FAIL(ALN),RAT(ALN)                  FI
C                                                                       FI
C        RETURN CONTROL TO THE MAIN PROGRAM                             FI
C                                                                       FI
      RETURN                                                            FI
C                                                                       FI
C        FORMAT STATEMENTS                                              FI
C                                                                       FI
C                                                                       FI
    5 FORMAT (1H1,3(T40,A10,2X,A10/T42,A10,2X,A10/)//T8,51HADJUSTED CUMU FI
     1LATIVE SUCCESS/FAILURE RATIO FOR LAST ,I4,23H FLIGHTS OF A TOTAL O FI
     2F ,I4//T10,13HLAUNCH NUMBER,T27,6HRESULT,T35,5HSTAGE,T47,6HTOTALS, FI
     3T58,8HPER CENT/T47,1HS,T52,1HF/)                                  FI
    6 FORMAT (T15,I4,T30,A1,T38,A1,T45,I4,T50,I4,T58,F6.2)              FI
    7 FORMAT (1H1,T8,43HADJUSTED SUCCESS/FAILURE RATIO FOR LAST ,I4,23H FI
     1FLIGHTS OF A TOTAL OF ,I4/T10,22HNUMBER OF SUCCESSES = ,I4/T10,21H FI
     2NUMBER OF FAILURES = ,I4/T10,14HRELIABILITY = ,F6.2,1H%//)        FI
      END                                                               FI
```

```
      SUBROUTINE FAILLOC (NO)
C
C        TYPE STATEMENTS
C
      INTEGER FL,FP,FA,FO,FAIL,ALN,BV,UV,SUCC
      INTEGER BV1,BV2,UV1,UV2
      REAL LRAT
C
C        COMMON BLOCKS
C
      COMMON /DATAIN/ ORGDAT,NEWDAT,IDATE(2000),BOOST(2000),UPPER(2000),
     1SOUR(2000),RES(2000),LOC(2000),PRODIR(2000),ST(2000,7),STG(2000),L
     2TYP(2000),MULTLA(2000),ALN
C
      COMMON /DATAOUT/ BV,UV,BV1,UV1,BV2,UV2,BVEH(20),UVEH(20),BCONT(20)
     1,UCONT(20),RESU,LATYP(6)
C
      COMMON /STORIT/ SUCC(1500),FAIL(1500),RAT(1500),HEAD(12)
C
C        ZERO THE FAILURE LOCATION COUNTERS
C
      FL=0
      FP=0
      FA=0
      FO=0
C
C        SET THE STARTING LAUNCH NUMBER
C
      NO1=NO*ALN/100.0
      IF (NO.EQ.0) NO1=ALN
      N1=ALN-NO1+1
C
C        THE FAILURES DURING EACH PHASE ARE NOW CALCULATED
C        IF THE PHASE OF FAILURE IS UNKNOWN THAT LAUNCH IS NOT INCLUDED
C
      DO 2 I=N1,ALN
      IF (STG(I).EQ.1HB.AND.BV.NE.0) GO TO 1
      IF (STG(I).EQ.1HU.AND.UV.NE.0) GO TO 1
      IF (BV.EQ.0.AND.UV.EQ.0.AND.RES(I).EQ.1HF) GO TO 1
      GO TO 2
    1 IF (LOC(I).EQ.1HP) FP=FP+1
      IF (LOC(I).EQ.1HL) FL=FL+1
      IF (LOC(I).EQ.1HA) FA=FA+1
      IF (LOC(I).EQ.1HO) FO=FO+1
    2 CONTINUE
C
C        CALCULATE THE TOTAL NUMBER OF FAILURES
C        IF NO FAILURES ARE FOUND, RETURN TO THE MAIN PROGRAM
C
      FAIL(1)=FP+FL+FA+FO
      IF (FAIL(1).EQ.0.0) WRITE (6,4)
      IF (FAIL(1).EQ.0.0) RETURN
C
C        CALCULATE THE PERCENTAGE OF FAILURES OCCURRING DURING EACH
C        PHASE
C
      PRAT=FP*100.0/FAIL(1)
```

```
      LRAT=FL*100.0/FAIL(1)
      ARAT=FA*100.0/FAIL(1)
      ORAT=FO*100.0/FAIL(1)
C
C     PRINT THE RESULTS
C
      WRITE (6,3) HEAD,NO1,ALN,FP,PRAT,FL,LRAT,FA,ARAT,FO,ORAT,FAIL(1)
C
C     RETURN CONTROL TO THE MAIN PROGRAM
C
      RETURN
C
C     FORMAT STATEMENTS
C
C
    3 FORMAT (1H1,3(T40,A10,2X,A10/T42,A10,2X,A10/)//T8,47HADJUSTED FAIL
     1URES CLASSED BY LOCATION FOR LAST ,I4,23H FLIGHTS OF A TOTAL OF ,I
     24//T12,8HLOCATION,T30,8HFAILURES,T48,8HPER CENT//T14,3HPAD,T32,I3,
     3T49,F6.2/T14,4HLAND,T32,I3,T49,F6.2/T14,6HASCENT,T32,I3,T49,F6.2/T
     414,7HORBITAL,T32,I3,T49,F6.2/T14,5HTOTAL,T31,I4,T49,5H100.00)
    4 FORMAT (///,T35,30HNO FAILURES HAVE BEEN REPORTED)
      END
```

```
      SUBROUTINE CURVIT (NO,C1,C2,C3,C4)                                  FI
C                                                                         FI
C        TYPE STATEMENTS                                                  FI
C                                                                         FI
      INTEGER C1,C2,C3,C4                                                 FI
      INTEGER ALN,SUCC,FAIL                                               FI
C                                                                         FI
C        COMMON BLOCKS                                                    FI
C                                                                         FI
      COMMON /DATAIN/ ORGDAT,NEWDAT,IDATE(2000),BOOST(2000),UPPER(2000),  FI
     1SUUR(2000),RES(2000),LOC(2000),PRODIR(2000),ST(2000,7),STS(2000),L  FI
     2TYP(2000),MULTLA(2000),ALN                                          FI
C                                                                         FI
      COMMON /STORIT/ SUCC(1500),FAIL(1500),RAT(1500),HEAD(12)            FI
C                                                                         FI
      COMMON /FIT/ X(2000),Y(2000),WF(2000),WORK(2500),THEOR(2000),NDATA  FI
     1,RELERR,ABSERR,IFLAG,PAR(6),R,NPAR                                  FI
C                                                                         FI
C        EXTERNAL THOSE SUBROUTINE NAMES WHICH ARE USED IN CALLS TO       FI
C        OTHER SUBROUTINES                                                FI
C                                                                         FI
C           THEOR1, THEOR2, THEOR3, THEOR4 CONTAIN THE EQUATIONS THAT     FI
C              ARE TO BE FIT                                              FI
C           DERIV1, DERIV2, DERIV3, DERIV4 CONTAIN THE FIRST AND SECOND   FI
C              PARTIAL DERIVATIVES OF THE CORRESPONDING THEOR SUBROUTINE  FI
C                                                                         FI
      EXTERNAL THEOR1,DERIV1,THEOR2,DERIV2,THEOR3,DERIV3,THEOR4,DERIV4    FI
C                                                                         FI
C        DATA BLOCK SETS THE VALUES OF THE ERROR TEST FOR THE CURVE       FI
C        FITTING SUBROUTINES                                              FI
C                                                                         FI
      DATA RELERR/1.E-4/,ABSERR/1.E-6/                                    FI
C                                                                         FI
C        DETERMINE THE NUMBER OF LAUNCHES TO BE CONSIDERED                FI
C                                                                         FI
      NO1=NO*ALN/100.0                                                    FI
C                                                                         FI
C        FILL THE *X* ARRAY WITH THE LAUNCH NUMBER (ADJUSTED TO THE NEW   FI
C        START)                                                           FI
C        FILL THE *Y* ARRAY WITH THE ACHIEVED RELIABILITY                 FI
C        FILL THE *WF* ARRAY WITH ONES SO ALL LAUNCHES ARE CONSIDERED     FI
C        EQUALLY                                                          FI
C                                                                         FI
      DO 1 I=1,NO1                                                        FI
      X(I)=I                                                              FI
      Y(I)=RAT(I+ALN-NO1)                                                 FI
      WF(I)=1.0                                                           FI
    1 CONTINUE                                                            FI
C                                                                         FI
C        SET *NDATA* = NUMBER OF LAUNCHES                                 FI
C        SET *NPAR* = NUMBER OF CURVE FIT PARAMETERS                      FI
C                                                                         FI
      NDATA=NO1                                                           FI
      NPAR=3                                                              FI
C                                                                         FI
C        TEST FOR FIT ON Y = A + B * EXP(C * X)                           FI
C                                                                         FI
```

```
      IF (C1.NE.1) GO TO 2
C
C     SET INITIAL GUESS ON PARAMETERS
C
      PAR(1)=1.0
      PAR(2)=-1.0
      PAR(3)=-0.5
      IFLAG=0
C
C     CALL CURVE FITTING SUBROUTINES
C
      CALL FITIT (THEOR1,DERIV1)
      IF (IFLAG.EQ.10) GO TO 2
C
C     WRITE EQUATION WITH CALCULATED PARAMETERS
C
      WRITE (6,6) PAR(1),PAR(2),PAR(3)
C
C     TEST FOR FIT ON Y = A * EXP(B * EXP(C * X))
C
    2 IF (C2.NE.1) GO TO 3
C
C     SET INITIAL GUESS ON PARAMETERS
C
      PAR(1)=1.0
      PAR(2)=-1.0
      PAR(3)=-1.0
      IFLAG=0
C
C     CALL CURVE FITTING SUBROUTINES
C
      CALL FITIT (THEOR2,DERIV2)
      IF (IFLAG.EQ.10) GO TO 3
C
C     WRITE EQUATION WITH CALCULATED PARAMETERS
C
      WRITE (6,7) PAR(1),PAR(2),PAR(3)
C
C     TEST FOR FIT ON Y = A * (1 - B/(C * X + 3))
C
    3 IF (C3.NE.1) GO TO 4
C
C     SET INITIAL GUESS ON PARAMETERS
C
      PAR(1)=1.0
      PAR(2)=1.0
      PAR(3)=1.0
      IFLAG=0
C
C     CALL CURVE FITTING SUBROUTINES
C
      CALL FITIT (THEOR3,DERIV3)
      IF (IFLAG.EQ.10) GO TO 4
C
C     WRITE EQUATION WITH CALCULATED PARAMETERS
C
      WRITE (6,8) PAR(1),PAR(2),PAR(3)
```

```
C                                                                        FI
C          TEST FOR FIT ON Y = A * EXP(B/X)                              FI
C                                                                        FI
    4 IF (C4.NE.1) GO TO 5                                               FI
C                                                                        FI
C          SET INITIAL GUESS ON PARAMETERS                               FI
C                                                                        FI
      PAR(1)=1.0                                                         FI
      PAR(2)=-0.5                                                        FI
      NPAR=2                                                             FI
      IFLAG=0                                                            FI
C                                                                        FI
C          CALL CURVE FITTING SUBROUTINES                                FI
C                                                                        FI
      CALL FITIT (THEOR4,DERIV4)                                         FI
      IF (IFLAG.EQ.10) GO TO 5                                           FI
C                                                                        FI
C          WRITE EQUATION WITH CALCULATED PARAMETERS                     FI
C                                                                        FI
      WRITE (6,9) PAR(1),PAR(2)                                          FI
    5 CONTINUE                                                           FI
C                                                                        FI
C          RETURN CONTROL TO MAIN PROGRAM                                FI
C                                                                        FI
      RETURN                                                             FI
C                                                                        FI
C          FORMAT STATEMENTS                                             FI
C                                                                        FI
C                                                                        FI
    6 FORMAT (1H1,43HTHE CURVE FIT FOR THIS DATA IS OF THE FORM://10X,44 FI
     1HRELIABILITY = A + B * EXP(C * LAUNCH NUMBER)//2X,39HWITH THE PARA FI
     2METERS A, B, C AS FOLLOWS://15X,3HA =,E18.10/15X,3HB =,E18.10/15X, FI
     33HC =,E18.10/)                                                     FI
    7 FORMAT (1H1,43HTHE CURVE FIT FOR THIS DATA IS OF THE FORM://10X,49 FI
     1HRELIABILITY = A * EXP(B * EXP(C * LAUNCH NUMBER))//2X,39HWITH THE FI
     2 PARAMETERS A, B, C AS FOLLOWS://15X,3HA =,E18.10/15X,3HB =,E18.10 FI
     3/15X,3HC =,E18.10/)                                                FI
    8 FORMAT (1H1,43HTHE CURVE FIT FOR THIS DATA IS OF THE FORM://10X,49 FI
     1HRELIABILITY = A * (1 - B/(C * LAUNCH NUMBER + 3))//2X,39HWITH THE FI
     2 PARAMETERS A, B, C AS FOLLOWS://15X,3HA =,E18.10/15X,3HB =,E18.10 FI
     3/15X,3HC =,E18.10/)                                                FI
    9 FORMAT (1H1,43HTHE CURVE FIT FOR THIS DATA IS OF THE FORM://10X,38 FI
     1HRELIABILITY = A * EXP(B/LAUNCH NUMBER)//2X,36HWITH THE PARAMETERS FI
     2 A, B AS FOLLOWS://15X,3HA =,E18.10/15X,3HB =,E18.10/)             FI
      END                                                                FI
```

```
      SUBROUTINE THEOR1                                                  F
      COMMON /FIT/ X(2000),Y(2000),WF(2000),WORK(2500),THEOR(2000),NDATA F
     1,RELERR,ABSERR,IFLAG,PAR(6),R,NPAR                                 F
      DO 2 I=1,NDATA                                                     F
C                                                                        F
C         CHECK FOR POSSIBLE EXPONENTIAL ARGUMENT OUT OF RANGE           F
C                                                                        F
      IF (PAR(3)*X(I).GE.-670.AND.PAR(3)*X(I).LE.735) GO TO 1            F
      WRITE (6,3)                                                        F
      IFLAG=1.                                                           F
      RETURN                                                             F
    1 THEOR(I)=PAR(1)+PAR(2)*EXP(PAR(3)*X(I))                            F
    2 CONTINUE                                                           F
      RETURN                                                             F
C                                                                        F
    3 FORMAT (1H1,45HEXPONENTIAL ARGUMENT TOO LARGE FOR CURVE FIT.//2X,4 F
     14HRELIABILITY = A + B * EXP(C * LAUNCH NUMBER)//)                  F
      END                                                                F
```

```
      SUBROUTINE DERIV1 (I,PF,PPF)                                          F
      DIMENSION PF(6), PPF(6,6)                                             F
      COMMON /FIT/ X(2000),Y(2000),WF(2000),WORK(2500),THEOR(2000),NDATA    F
     1,RELERR,ABSERR,IFLAG,PAR(6),R,NPAR                                    F
      PF(1)=1.                                                              F
      PF(2)=EXP(PAR(3)*X(I))                                                F
      PF(3)=X(I)*PAR(2)*PF(2)                                               F
      PPF(1,1)=0.0                                                          F
      PPF(2,1)=0.0                                                          F
      PPF(3,1)=0.0                                                          F
      PPF(1,2)=PPF(2,1)                                                     F
      PPF(2,2)=0.0                                                          F
      PPF(3,2)=X(I)*PF(2)                                                   F
      PPF(1,3)=PPF(3,1)                                                     F
      PPF(2,3)=PPF(3,2)                                                     F
      PPF(3,3)=X(I)*PF(3)                                                   F
      RETURN                                                                F
      END                                                                   F
```

```
      SUBROUTINE THEOR2                                                 FI
      COMMON /FIT/ X(2000),Y(2000),WF(2000),WORK(25000),THEOR(2000),NDATA FI
     1,RELERR,ABSERR,IFLAG,PAR(6),R,NPAR                                FI
      DO 2 I=1,NDATA                                                    FI
C                                                                       FI
C        CHECK FOR POSSIBLE EXPONENTIAL ARGUMENT OUT OF RANGE           FI
C                                                                       FI
      IF (EXP(PAR(3)*X(I)).GE.-670.AND.EXP(PAR(3)*X(I)).LE.735) GO TO 1 FI
      WRITE (6,3)                                                       FI
      IFLAG=13                                                          FI
      RETURN                                                            FI
    1 THEOR(I)=PAR(1)*EXP(PAR(2)*EXP(PAR(3)*X(I)))                      FI
    2 CONTINUE                                                          FI
      RETURN                                                            FI
C                                                                       FI
    3 FORMAT (1H1,45HEXPONENTIAL ARGUMENT TOO LARGE FOR CURVE FITS//2X,4 FI
     19HRELIABILITY = A * EXP(B * EXP(C * LAUNCH NUMBER))//)            FI
      END                                                               FI
```

```
      SUBROUTINE DERIV2 (I,PF,PPF)                                       F
      DIMENSION PF(6), PPF(6,6)                                          F
      COMMON /FIT/ X(2000),Y(2000),WF(2000),WORK(2500),THEOR(2000),NDATA F
     1,RELERR,ABSERR,IFLAG,PAR(6),R,NPAR                                 F
      A=EXP(PAR(3)*X(I))                                                 F
      PF(1)=EXP(PAR(2)*A)                                                F
      PF(2)=PAR(1)*EXP(PAR(3)*X(I)+PAR(2)*A)                             F
      PF(3)=PAR(1)*PAR(2)*X(I)*EXP(PAR(3)*X(I)+PAR(2)*A)                 F
      PPF(1,1)=0.0                                                       F
      PPF(2,1)=PF(2)/PAR(1)                                              F
      PPF(3,1)=PF(3)/PAR(1)                                              F
      PPF(1,2)=PPF(2,1)                                                  F
      PPF(2,2)=PF(2)*A                                                   F
      PPF(3,2)=PF(2)*X(I)*(1+PAR(2)*A)                                   F
      PPF(1,3)=PPF(3,1)                                                  F
      PPF(2,3)=PPF(3,2)                                                  F
      PPF(3,3)=PF(3)*X(I)*(1+PAR(2)*A)                                   F
      RETURN                                                             F
      END                                                                F
```

```
      SUBROUTINE THEOR3                                                    F
      COMMON /FIT/ X(2000),Y(2000),WF(2000),WORK(2500),THEOR(2000),NDATA   F
     1,RELERR,ABSERR,IFLAG,PAR(6),R,NPAR                                   F
      DO 1 I=1,NDATA                                                       F
      THEOR(I)=PAR(1)*(1-PAR(2)/(PAR(3)*X(I)+PAR(2)))                      F
    1 CONTINUE                                                             F
      RETURN                                                               F
      END                                                                  F
```

```
      SUBROUTINE DERIV3 (I,PF,PPF)                                            FI
      DIMENSION PF(6), PPF(6,6)                                               FI
      COMMON /FIT/ X(2000),Y(2000),WF(2000),WORK(2500),THEOR(2000),NDATA      FI
     1,RELERR,ABSERR,IFLAG,PAR(6),R,NPAR                                      FI
      A=PAR(3)*X(I)+PAR(2)                                                    FI
      PF(1)=1-PAR(2)/A                                                        FI
      PF(2)=-PAR(1)*PAR(3)*X(I)/A**2.0                                        FI
      PF(3)=PAR(1)*PAR(2)*X(I)/A**2.0                                         FI
      PPF(1,1)=0.0                                                            FI
      PPF(2,1)=PF(2)/PAR(1)                                                   FI
      PPF(3,1)=PF(3)/PAR(1)                                                   FI
      PPF(1,2)=PPF(2,1)                                                       FI
      PPF(2,2)=-2.0*PF(2)/A                                                   FI
      PPF(3,2)=-(PF(2)*X(I)+PF(3))/A                                          FI
      PPF(1,3)=PPF(3,1)                                                       FI
      PPF(2,3)=PPF(3,2)                                                       FI
      PPF(3,3)=-2.0*X(I)*PF(3)/A                                              FI
      RETURN                                                                  FI
      END                                                                     FI
```

```
      SUBROUTINE THEOR4                                                        F
      COMMON /FIT/ X(2000),Y(2000),WF(2000),WORK(2500),THEOR(2000),NDATA F
     1,RELERR,ABSERR,IFLAG,PAR(6),R,NPAR                                       F
      DO 2 I=1,NDATA                                                           F
C                                                                              F
C        CHECK FOR POSSIBLE EXPONENTIAL ARGUMENT OUT OF RANGE                  F
C                                                                              F
      IF (PAR(2)/X(I).GE.-676.AND.PAR(2)/X(I).LE.735) GO TO 1                  F
      WRITE (6,3)                                                              F
      IFLAG=10                                                                 F
      RETURN                                                                   F
    1 THEOR(I)=PAR(1)*EXP(PAR(2)/X(I))                                         F
    2 CONTINUE                                                                 F
      RETURN                                                                   F
C                                                                              F
    3 FORMAT (1H1,45HEXPONENTIAL ARGUMENT TOO LARGE FOR CURVE FITS///2X,3 F
     18HRELIABILITY = A * EXP(B/LAUNCH NUMBER)//)                               F
      END                                                                      F
```

```
      SUBROUTINE DERIV4 (I,PF,PPF)                                      FI
      DIMENSION PF(6), PPF(6,6)                                         FI
      COMMON /FIT/ X(2000),Y(2000),WF(2000),WORK(2500),THEOR(2000),NDATA FI
     1,RELERR,ABSERR,IFLAG,PAR(6),R,NPAR                                FI
      PF(1)=EXP(PAR(2)/X(I))                                            FI
      PF(2)=PAR(1)*EXP(PAR(2)/X(I))/X(I)                                FI
      PPF(1,1)=0.0                                                      FI
      PPF(2,1)=PF(1)/X(I)                                               FI
      PPF(1,2)=PPF(2,1)                                                 FI
      PPF(2,2)=PF(2)/X(I)                                               FI
      RETURN                                                            FI
      END                                                               FI
```

```
      SUBROUTINE FITIT (THEORY,DERIV)                                    F
C                                                                        F
C                         AFWL SCIENTIFIC PROGRAM LIBRARY               F
C     AFWL IDENTIFICATION - FITIT                                        F
C     AFWL CONTACT - TECHNOLOGY DIVISION,THEORETICAL BRANCH,             F
C                    MATHEMATICS SECTION, EXT. 9451                      F
C     DATE ESTABLISHED IN LIBRARY -                                      F
C     DATE OF LAST MODIFICATION   -                                      F
C                                                                        F
C                                                                        F
C     PROGRAMMER - CAPT. JAMES H. HEAD                                   F
C                  USAF ACADEMY                                          F
C                  COLORADO 80840                                        F
C                  MODIFIED FOR AFWL USE BY LT. HENRY J. HAPP            F
C                                                                        F
C                                                                        F
C     DESCRIPTION OF SUBROUTINE CALLING ARGUMENTS                        F
C                                                                        F
C     NDATA...INPUT...NUMBER OF DATA POINTS IN THE X AND Y ARRAYS TO     F
C             BE FIT.                                                    F
C     NPAR....INPUT...NUMBER OF PARAMETERS IN THE PAR ARRAY.             F
C     THEORY..EXTERNAL...SUBROUTINE WHCIH DEFINES THE THEORETICAL FIT    F
C             VALUES AT THE GIVEN X VALUES.                              F
C     DERIV...EXTERNAL...SUBROUTINE WHICH DEFINES THE FIRST AND SECOND   F
C             PARTIAL DERIVATIVES OF THE FIT EXPRESSION WITH RESPECT     F
C             TO THE PARAMETERS AT EACH OF THE DATA POINTS.              F
C     X.......INPUT...ARRAY OF ABSCISSA DATA VALUES.                     F
C     Y.......INPUT...ARRAY OF ORDINATE DATA VALUES.                     F
C     WF......INPUT...ARRAY OF POSITIVE WEIGHTING FACTORS FOR THE DATA.  F
C     RELERR..INPUT...RELATIVE ERROR TOLERANCE FOR PARAMETER CONVERGENCE F
C     ABSERR..INPUT...ABSOLUTE ERROR TOLERANCE FOR PARAMETER CONVERGENCE F
C     IFLAG...INPUT/OUTPUT...ON INPUT, IFLAG=-1 WILL PRINT INTERMEDIATE  F
C             ITERATION VALUES AS THE SUBROUTINE CONVERGES, AS WELL AS   F
C             STATEMENTS GIVING THE REASONS FOR NON-CONVERGENCE.  ON     F
C             OUTPUT, IFLAG IS AN ERROR INDICATION FLAG.                 F
C             =0  CONVERGENCE.  NORMAL RETURN.                           F
C             =1  CONVERGENCE.  RESIDUALS ARE ZERO.  POSSIBLE LIMITING   F
C                 PRECISION EFFECTS.                                     F
C             =2  SLOW CONVERGENCE.  RESIDUAL IS GREATER THAN 0.9 TIMES  F
C                 THE PREVIOUS RESIDUAL FOR NSCS ITERATIONS.             F
C             =3  MAXIMUM NUMBER OF ITERATIONS EXCEEDED.                 F
C             =4  POSSIBLE DIVERGENCE.  RESIDUAL HAS REMAINED LARGER     F
C                 THAN 10 TIMES THE SMALLEST RESIDUAL FOR NCDS           F
C                 ITERATIONS.                                            F
C             =5  POSSIBLE DIVERGENCE.  LARGEST SOLUTION INCREMENT       F
C                 NORM HAS INCREASED BY A FACTOR OF 10 FOR THE LAST      F
C                 NCD2S ITERATIONS.                                      F
C             =6  POSSIBLE LOCAL MINIMUM.  MAXIMUM NUMBER OF CUT STEP    F
C                 ITERATIONS TAKEN.                                      F
C             =7  A MATRIX IS SINGULAR.  THE FORMULATION OF THE THEORY   F
C                 AND/OR DERIV SUBROUTINES MAY BE INCORRECT.             F
C     PAR.....INPUT/OUTPUT...ARRAY OF PARAMETER VALUES.  ON INPUT, PAR   F
C             CONTAINS AN INITIAL ESTIMATE OF THE PARAMETER VALUES.  ON  F
C             OUTPUT, PAR CONTAINS THE LAST VALUE OBTAINED BY THE        F
C             SUBROUTINE AS IT ITERATED.                                 F
C     RES.....OUTPUT...RESIDUAL VALUE.                                   F
C     WORK....SCRATCH...WORK ARRAY.  WORK MUST BE DIMENSIONED AT LEAST   F
```

```
C           2*NPAR**2 + 3*NPAR + NDATA.                                 FI
C                                                                       FI
C                                                                       FI
C     ABSTRACT                                                          FI
C                                                                       FI
C     SUBROUTINE FITIT FITS A USER-PRESCRIBED FUNCTION OF ONE VARIABLE  FI
C     AND  NPAR  PARAMETERS TO A SET OF DISCRETE DATA POINTS.  THE      FI
C     FIT IS A LEAST-SQUARES FIT, I.E., THE SUM OF THE SQUARES OF THE   FI
C     RESIDUALS IS MINIMIZED.                                           FI
C                                                                       FI
C     THE FIT IS ACTUALLY ACCOMPLISHED BY SUBROUTINE  FIT1.  FITIT      FI
C     ALLOCATES VIRTUAL STORAGE IN THE ARRAY  WORK  AND CALLS  FIT1.    FI
C     THIS ELIMINATES THE NEED FOR A LONG CALL LIST AND ALLOWS THE      FI
C     NUMBER OF DATA POINTS AND PARAMETERS TO REMAIN ARBITRARY.         FI
C                                                                       FI
C     FITIT  ACCOMPLISHES THE FIT BY TRUNCATING THE TAYLOR SERIES FOR   FI
C     EACH FITTING PARAMETER ABOUT THE INITIAL APPROXIMATION AFTER THE  FI
C     QUADRATIC TERM, AND USING THIS NEW VALUE TO REPLACE THE INITIAL   FI
C     VALUE, ITERATES THIS PROCEDURE UNTIL THE NUMBER OF SIGNIFICANT    FI
C     DIGITS DESIRED IS OBTAINED, OR UNTIL THE MAXIMUM NUMBER OF        FI
C     ITERATIONS ALLOWED (NITMAX) IS REACHED.                           FI
C                                                                       FI
C     THE BASIC CODE IS EXPLAINED IN  *FITIT, A PROGRAM TO LEAST-SQUARES FI
C     FIT NON-LINEAR THEORIES* BY JAMES M. MEAD, LIBRARY OF CONGRESS    FI
C     CATALOG NUMBER *AFA TR 70-5*.                                     FI
C                                                                       FI
C     THE USER MUST SUPPLY TWO(2) SUBROUTINES NAMED  THEORY  AND  DERIV. FI
C     THEORY  MUST TAKE THE FOLLOWING FORM -                            FI
C                                                                       FI
C           THEORY(NDATA,PAR,X,THEOR)                                   FI
C                                                                       FI
C           WHERE  NDATA  IS THE NUMBER OF DATA POINTS,  PAR  IS THE    FI
C           CURRENT VECTOR OF FITTING PARAMETERS,  X  IS THE ARRAY OF   FI
C           DATA ABSCISSAE, AND  THEOR  IS THE VECTOR OF PREDICTED      FI
C           THEORETICAL VALUES.  THAT IS,                               FI
C                                                                       FI
C           THEOR(I) = THEORY(X(I),PAR),  I=1,....,NDATA                FI
C                                                                       FI
C     DERIV  MUST TAKE THE FOLLOWING FORM -                             FI
C                                                                       FI
C           DERIV(I,NPAR,X,PAR,PF,PPF)                                  FI
C                                                                       FI
C           WHERE  I  IS THE INDEX OF THE DATA POINT,  NPAR  IS THE     FI
C           NUMBER OF FITTING PARAMETERS,  X  AND  PAR  ARE AS IN       FI
C           THEORY,  PF  IS THE VECTOR OF FIRST PARTIAL DERIVATIVES,    FI
C           AND  PPF  IS THE MATRIX OF SECOND PARTIAL DERIVATIVES.      FI
C           THAT IS,                                                    FI
C                                                                       FI
C           PF(J) = D(THEORY(X(I),PAR))/D(PAR(J)),  J=1,....,NPAR       FI
C                                                                       FI
C                       2                                               FI
C           PPF(J,K) = D (THEORY(X(I),PAR))/D(PAR(J))D(PAR(K))          FI
C                                          J=1,....,NPAR, K=1,....,NPAR FI
C                                                                       FI
C     THE SCRATCH ARRAY  WORK  ALLOCATES VIRTUAL STORAGE AND MUST BE    FI
C     DIMENSIONED                                                       FI
C           WORK(2*NPAR**2 + 3*NPAR + NDATA)                            FI
```

```
C                                                                       FI
C                                                                       FI
      COMMON /FIT/ X(2000),Y(2000),WF(2000),WORK(2500),THEOR(2000),NDATA FI
     1,RELERR,ABSERR,IFLAG,PAR(6),R,NPAR                                FI
      EQUIVALENCE (R,RES)                                               FI
      EXTERNAL THEORY,DERIV                                             FI
C                                                                       FI
C     SET MAXIMUMS                                                      FI
C                                                                       FI
      DATA NITMAX/500/                                                  FI
      DATA NCSTP/4/                                                     FI
      DATA NCCSI/10/                                                    FI
      DATA NSCS/50/                                                     FI
      DATA NCOS/10/                                                     FI
      DATA NCO2S/4/                                                     FI
C                                                                       FI
C     SET UP ACCOUNTING PROCEDURE                                       FI
C                                                                       FI
      DATA KKKKKK/0/                                                    FI
      IF (KKKKKK.EQ.0) CALL REMARK (11H* * * FITIT)                     FI
      KKKKKK=1                                                          FI
C                                                                       FI
C     CHECK INPUT PARAMETERS                                            FI
C                                                                       FI
      IF (NPAR.LT.1) GO TO 11                                           FI
      IF (NDATA.LE.NPAR) GO TO 12                                       FI
      DO 2 I=1,NDATA                                                    FI
      IF (WF(I).LE.0.0) GO TO 13                                        FI
      DO 1 J=1,NDATA                                                    FI
      IF (J.EQ.I) GO TO 1                                               FI
      IF (X(J).EQ.X(I)) GO TO 14                                        FI
    1 CONTINUE                                                          FI
    2 CONTINUE                                                          FI
      IF (ABSERR.LT.0.0) GO TO 15                                       FI
      IF (RELERR.LT.0.0) GO TO 15                                       FI
      IF (ABSERR+RELERR.EQ.0.0) GO TO 15                                FI
C                                                                       FI
C     SET INDICES FOR VIRTUAL STORAGE ALLOCATION                        FI
C                                                                       FI
      NPAR2=36                                                          FI
      IPF=1                                                             FI
      IPPF=IPF+6                                                        FI
      ITHEOR=IPPF+NPAR2                                                 FI
      IC=ITHEOR+NDATA                                                   FI
      IA=IC+6                                                           FI
      IOLD=IA+NPAR2                                                     FI
C                                                                       FI
C     SET IPRN                                                          FI
C                                                                       FI
      IPRIN=ISIGN(1,IFLAG)                                              FI
C                                                                       FI
C     FIT THE DATA                                                      FI
C                                                                       FI
      CALL FITI (THEORY,DERIV,NITMAX,NCSTP,NCCSI,NSCS,NCOS,NCO2S,IPRIN) FI
C                                                                       FI
C     CHECK PRINT AND ERROR FLAGS                                       FI
C                                                                       FI
```

```
      IF (IFLAG.EQ.10) RETURN                                          F1
      IF (IFLAG.EQ.7) GO TO 16                                         F1
      IF (IPRIN.NE.-1) GO TO 10                                        F1
      IFL=IFLAG+1                                                      F1
      GO TO (3,4,5,6,7,8,9), IFL                                       F1
C                                                                      F1
    3 PRINT 18, IFLAG                                                  F1
      GO TO 10                                                         F1
    4 PRINT 19, IFLAG                                                  F1
      GO TO 10                                                         F1
    5 PRINT 20, NSGS,IFLAG                                             F1
      GO TO 10                                                         F1
    6 PRINT 21, NITMAX,IFLAG                                           F1
      GO TO 10                                                         F1
    7 PRINT 22, NCDS,IFLAG                                             F1
      GO TO 10                                                         F1
    8 PRINT 22, NCD2S,IFLAG                                            F1
      GO TO 10                                                         F1
    9 PRINT 23, NCGSI,IFLAG                                            F1
C                                                                      F1
   10 CONTINUE                                                         F1
      RETURN                                                           F1
C                                                                      F1
   11 PRINT 24, NPAR                                                   F1
      GO TO 17                                                         F1
   12 PRINT 25, NDATA,NPAR                                             F1
      GO TO 17                                                         F1
   13 PRINT 26, I,WF(I)                                                F1
      GO TO 17                                                         F1
   14 PRINT 27, I,J,X(I)                                               F1
      GO TO 17                                                         F1
   15 PRINT 28, ABSERR,RELERR                                          F1
      GO TO 17                                                         F1
   16 PRINT 29                                                         F1
C                                                                      F1
      RETURN                                                           F1
   17 IFLAG=10                                                         F1
C                                                                      F1
   18 FORMAT (//,64H CONGRATULATIONS.  THE SUBROUTINE CONVERGED NORMALLY F1
     1 WITH IFLAG=,I1,1H.)                                             F1
   19 FORMAT (//115H THE SUBROUTINE CONVERGED SUCH THAT ALL RESIDUALS AR F1
     1E ZERO.  THERE ARE POSSIBLE LIMITING PRECISION EFFECTS.  IFLAG=,I1 F1
     2,1H.)                                                            F1
   20 FORMAT (//82H SUBROUTINE IS CONVERGING VERY SLOWLY.  RESIDUAL IS . F1
     1GT. 0.9*PREVIOUS RESIDUAL FOR,I4,15H STEPS.  IFLAG=,I1,1H.)      F1
   21 FORMAT (//,37H SUBROUTINE WAS UNABLE TO CONVERGE IN,I4,75H ITERATI F1
     1ONS.  PERHAPS LESS STRINGENT ERROR TOLERANCES ARE REQUIRED.  IFLAG F1
     2=,I1,1H.)                                                        F1
   22 FORMAT (//,104H POSSIBLE DIVERGENCE.  LARGEST SOLUTION INCREMENT N F1
     1ORM HAS INCREASED BY A FACTOR OF AT LEAST 10 FOR THE LAST,I4,20H I F1
     2TERATIONS.  IFLAG=,I1,1H.)                                       F1
   23 FORMAT (//,24H POSSIBLE LOCAL MINIMUM.,I5,55H CUT STEP ITERATIONS  F1
     1TAKEN WITH NO SOLUTION IMPROVEMENT.  IFLAG =,I2,1H.)             F1
   24 FORMAT (59H1THE NUMBER OF PARAMETERS MUST BE .GE. 1.  AS INPUT, NP F1
     1AR =,I4,1H.)                                                     F1
   25 FORMAT (106H1TO COMPUTE A NONLINEAR LEAST SQUARES FIT, THE NUMBER  F1
     1OF DATA POINTS MUST EXCEED THE NUMBER OF PARAMETERS.//18H AS INPUT F1
```

```
      2, NDATA =,I5,11H AND NPAR =,I4,1H.)                             FI
   26 FORMAT (75H1THE WEIGHTING FACTORS FOR THE DATA VALUES MUST BE POSI FI
     1TIVE.  AS INPUT, WF(,I4,3H) =,1PG11.3)                           FI
   27 FORMAT (51H1NO TWO ABSCISSA VALUES MAY BE EQUAL.  AS INPUT, X(,I4, FI
     16H) = X(,I4,3H) =,1PG11.3,1H.)                                   FI
   28 FORMAT (106H1THE ERROR TOLERANCES RELERR AND ABSERR MUST BE NON-NE FI
     1GATIVE.  IN ADDITION, AT LEAST ONE MUST BE POSITIVE.//19H AS INPUT FI
     2, ABSERR =,1PG11.3,13H AND RELERR =,1PG11.3,1H.)                 FI
   29 FORMAT (47H1THE *A* MATRIX IN SUBROUTINE FITI IS SINGULAR.,/,132H0 FI
     1POSSIBLE CAUSES ARE AN INCORRECT FORMULATION OF THE THEORY AND/OR  FI
     2DERIV SUBROUTINES, OR A POOR INITIAL ESTIMATE FOR THE PARAMETERS.) FI
      END                                                              FI
```

```
      SUBROUTINE FITI (THEORY,DERIV,NITMAX,NCSTP,NCCSI,NSCS,NCDS,NCD2S,I   FI
     1PRIN)                                                                FI
C                                                                          FI
C     ******************************************************************* FI
C                                                                        * FI
C     FITIT MERELY ALLOCATES VIRTUAL STORAGE FOR SUBROUTINE FITI.  FITI   FI
C     PERFORMS THE ACTUAL WORK.                                           FI
C                                                                        * FI
C     EXPLANATION OF SUBROUTINE ARGUMENTS                                 FI
C                                                                        * FI
C     NDATA......INPUT...NUMBER OF DATA POINTS IN THE X AND Y ARRAYS TO B FI
C             FIT.                                                       * FI
C     NPAR.......INPUT...NUMBER OF PARAMETERS IN THE PAR ARRAY.           FI
C     THEORY.....EXTERNAL...SUBROUTINE WHICH DEFINES THE THEORETICAL FIT  FI
C             VALUES AT THE GIVEN ABSCISSA (X) VALUES.                   * FI
C     DERIV......EXTERNAL...SUBROUTINE WHICH DEFINES THE FIRST AND SECOND FI
C             PARTIAL DERIVATIVES OF THE FIT EXPRESSION WITH RESPECT TO  * FI
C             THE PARAMETERS AT EACH OF THE DATA POINTS.                 * FI
C     X..........INPUT...ARRAY OF ABSCISSA DATA VALUES.                   FI
C     Y..........INPUT...ARRAY OF ORDINATE DATA VALUES.                   FI
C     WF.........INPUT...ARRAY OF POSITIVE WEIGHTING FACTORS FOR THE DATA FI
C     RELERR.....INPUT...RELATIVE ERROR TOLERANCE FOR PARAMETER CONVERGEN FI
C     ABSERR.....INPUT...ABSOLUTE ERROR TOLERANCE FOR PARAMETER CONVERGEN FI
C     NITMAX.....INPUT...MAXIMUM NUMBER OF ITERATIONS ALLOWED.            FI
C     NCSTP......INPUT...MAXIMUM NUMBER OF CONSECUTIVE CUT STEPS ALLOWED  FI
C             CUT STEP ITERATION.                                         FI
C     NCCSI......INPUT...MAXIMUM NUMBER OF CUT STEP ITERATIONS ALLOWED PE FI
C             ITERATION.                                                 * FI
C     NSCS.......INPUT...MAXIMUM ALLOWABLE STEPS WHERE THE ITERATION IS   FI
C             SLOWLY CONVERGENT.                                         * FI
C     NCDS.......INPUT...MAXIMUM ALLOWABLE CONSECUTIVE DIVERGENT STEPS.   FI
C     NCD2S......INPUT...MAXIMUM NUMBER OF STEPS WHERE SOLUTION INCREMENT FI
C             NORM MAY INCREASE BY A FACTOR OF 10 OVER THE PREVIOUS      * FI
C             ITERATION.                                                 * FI
C     IPRIN......INPUT...PRINT FLAG.  INTERMEDIATE RESULTS ARE PRINTED IF FI
C             IPRIN = +1.                                                * FI
C     PAR........INPUT/OUTPUT...ARRAY OF PARAMETER VALUES.  ON INPUT, PAR FI
C             CONTAINS THE INITIAL ESTIMATE.  ON OUTPUT, PAR CONTAINS THE* FI
C             LATEST MACHINE ITERATION RESULT.                           * FI
C     RES........OUTPUT...NORM OF THE RESIDUALS AFTER COMPUTATION.        FI
C     IFLAG......OUTPUT...ERROR INDICATION FLAG.                          FI
C             =0   CONVERGENCE.  NORMAL RETURN.                          * FI
C             =1   CONVERGENCE. RESIDUALS ARE ZERO.  POSSIBLE LIMITING   * FI
C                  PRECISION EFFECTS.                                    * FI
C             =2   SLOW CONVERGENCE.  RESIDUAL IS GREATER THAN 0.9 TIMES * F
C                  THE PREVIOUS RESIDUAL FOR NSCS ITERATIONS.            * FI
C             =3   MAXIMUM NUMBER OF ITERATIONS EXCEEDED.                * FI
C             =4   POSSIBLE DIVERGENCE.  RESIDUAL HAS REMAINED LARGER    * FI
C                  THAN 10 TIMES THE SMALLEST RESIDUAL FOR NCDS          * FI
C                  ITERATIONS.                                           * FI
C             =5   POSSIBLE DIVERGENCE.  LARGEST SOLUTION INCREMENT NORM * FI
C                  HAS INCREASED BY A FACTOR OF 10 FOR THE LAST NCD2S    * FI
C                  ITERATIONS.                                           * FI
C             =6   POSSIBLE LOCAL MINIMUM.  MAXIMUM NUMBER OF CUT STEP   * FI
C                  ITERATIONS TAKEN.                                     * FI
C             =7   A MATRIX IS SINGULAR.  THE FORMULATION OF THE THEORY  * FI
C                  AND/OR DERIV SUBROUTINES MAY BE INCORRECT.            * FI
```

```
C     THE REST OF THE ARGUMENTS ARE WORK ARRAYS.
C     **********************************************************************
C
      DIMENSION PF(6), PPF(6,6), C(6), OLD(6), A(6,6)
      COMMON /FIT/ X(2000),Y(2000),WF(2000),WORK(2500),THEOR(2000),NDATA
     1,RELERR,ABSERR,IFLAG,PAR(6),R,NPAR
      EQUIVALENCE (R,RES)
C
      DATA URO/7.E-15/
      DATA XMAX/37767777777777777777B/
C
C     COMPUTER UNIT ROUND-OFF VALUE
C     LARGEST POSITIVE FLOATING POINT NUMBER
C
C
C     SET INITIAL VALUES
C
      RE=AMAX1(RELERR,URO)
      PREV=XMAX
      ZNS=0.0
      NCUT=0
      ID=0
      ID3=0
      N1=NCOS*NCSTP
C
      RN=0.0
      DO 1 I=1,NDATA
      RN=RN+WF(I)
    1 CONTINUE
      RN=1./RN
C
      IF (IPRIN.EQ.-1) PRINT 33
C
C     *****  BEGIN ITERATION  *****
C
      DO 23 M=1,NITMAX
      KONV=1
      MM1=M-1
C
C     BEGIN CUT STEP LOOP
C
      DO 7 NCUT=1,NCSTP
      RES=0.0
      CALL THEORY
      IF (IFLAG.EQ.13) RETURN
      DO 2 I=1,NDATA
      THEOR(I)=THEOR(I)-Y(I)
      RES=RES+THEOR(I)*THEOR(I)*WF(I)
    2 CONTINUE
      RES=RES*RN
C
C     PRINT INTERMEDIATE SOLUTION VALUES, IF DESIRED
C
      IF (IPRIN.NE.-1) GO TO 3
      NN1=NCUT-1
      PRINT 32, MM1,NN1,RES,(PAR(I),I=1,NPAR)
C
```

```
C     CHECK RESIDUAL AND ITERATION PARAMETERS                          FI
C                                                                      FI
    3 CONTINUE                                                         FI
      IF (RES.EQ.0.0) GO TO 24                                         FI
      IF (RES.LT.PREV) GO TO 8                                         FI
C                                                                      FI
C     CHECK FOR POSSIBLE DIVERGENCE                                    FI
C                                                                      FI
      IF (RES.LT.10.*PREV) GO TO 4                                     FI
      ID3=ID3+1                                                        FI
      IF (ID3.LT.N1) GO TO 5                                           FI
      GO TO 26                                                         FI
C                                                                      FI
    4 CONTINUE                                                         FI
      ID3=0                                                            FI
C                                                                      FI
C     NEW PARAMETER VALUES DID NOT IMPROVE FIT.  TAKE AVERAGE OF OLD AND FI
C     NEW PARAMETERS AND TRY AGAIN.                                    FI
C                                                                      FI
    5 CONTINUE                                                         FI
      DO 6 I=1,NPAR                                                    FI
      PAR(I)=(OLD(I)+PAR(I))*0.5                                       FI
    6 CONTINUE                                                         FI
C                                                                      FI
C     END OF CUT STEP LOOP                                             FI
C                                                                      FI
    7 CONTINUE                                                         FI
C                                                                      FI
C     MAXIMUM NUMBER OF CONSECUTIVE CUT STEPS EXCEEDED.  INCREMENT AND FI
C     CHECK CUT STEP ITERATION COUNTER.                                FI
C                                                                      FI
      ID=ID+1                                                          FI
      IF (ID.LT.NCCSI) GO TO 12                                        FI
      GO TO 28                                                         FI
C                                                                      FI
C     TEST FOR SLOW CONVERGENCE                                        FI
C                                                                      FI
    8 CONTINUE                                                         FI
      IF (RES.LT.0.9*PREV) GO TO 9                                     FI
      ISC=ISC+1                                                        FI
      IF (ISC.LT.NSCS) GO TO 10                                        FI
      GO TO 25                                                         FI
C                                                                      FI
C     NEW VALUES IMPROVED FIT.  RESET ABNORMAL TERMINATION COUNTERS AND FI
C     SAVE NEW VALUES                                                  FI
C                                                                      FI
    9 CONTINUE                                                         FI
      ISC=0                                                            FI
C                                                                      FI
   10 CONTINUE                                                         FI
      ID=0                                                             FI
      ID2=0                                                            FI
      DO 11 I=1,NPAR                                                   FI
      OLD(I)=PAR(I)                                                    FI
   11 CONTINUE                                                         FI
C                                                                      FI
C       INITIALIZE WORK VARIABLES                                      FI
```

```
C
   12 CONTINUE
      DO 14 I=1,NPAR
      DO 13 J=1,NPAR
      A(J,I)=0.0
   13 CONTINUE
      C(I)=0.0
   14 CONTINUE
C
C     SET UP VARIABLES TO CALCULATE TAYLOR SERIES EXPANSION
C
      DO 17 K=1,NDATA
      CALL DERIV (K,PF,PPF)
      T=THEOR(K)
      U=WF(K)
      G=T*U
      DO 16 J=1,NPAR
      F=PF(J)
      C(J)=C(J)+G*F
      DO 15 I=1,NPAR
      A(I,J)=A(I,J)+U*(T*PPF(I,J)+PF(I)*F)
   15 CONTINUE
   16 CONTINUE
   17 CONTINUE
C
C     INVERT THE A MATRIX.  CHECK FOR SINGULARITY
C
      CALL INVRT (A,PF)
      IF (IFLAG.EQ.7) GO TO 29
C
C     CALCULATE NEW PARAMETER VALUES
C
      DO 19 I=1,NPAR
      Z=0.0
      ZNORM=0.0
      DO 18 J=1,NPAR
      Z=Z+A(I,J)*C(J)
   18 CONTINUE
      ZNORM=AMAX1(ZNORM,ABS(Z))
      PAR(I)=PAR(I)-Z
C
C     CHECK FOR CONVERGENCE
C
      IF (ABS(Z).GT.(RE*ABS(OLD(I))+ABSERR)) KONV=0
   19 CONTINUE
      IF (KONV.EQ.0) GO TO 20
C
C     PROCEDURE CONVERGED TO A SOLUTION
C
      IFLAG=0
      GO TO 30
C
C     NO CONVERGENCE.  TEST FOR DIVERGENCE
C
   20 CONTINUE
      IF (MM1.EQ.0) GO TO 21
      IF (ZNORM.LE.10.*ZNS) GO TO 21
```

```
      ID2=ID2+1
      IF (ID2.LT.NCD2S) GO TO 22
      GO TO 27
C
   21 CONTINUE
      ID2=0
C
C     SET VARIABLES TO BE SAVED AND ITERATE AGAIN
C
   22 CONTINUE
      PREV=AMIN1(PREV,RES)
      ZNS=ZNORM
   23 CONTINUE
C
C     *****  END OF ITERATION  *****
C
C
C     MAXIMUM NUMBER OF ITERATIONS EXCEEDED.  ITERATION MAY BE
C     CONVERGING VERY SLOWLY
      IFLAG=3
      GO TO 30
C
C     RESIDUALS ZERO.  PROBABLE CONVERGENCE WITH POSSIBLE LIMITING
C     PRECISION
C
   24 IFLAG=1
      GO TO 30
C
C     SLOW CONVERGENCE.  RESIDUAL IS GREATER THAN .9 TIMES THE PREVIOUS
C     RESIDUAL FOR NSCS ITERATIONS
   25 IFLAG=2
      GO TO 30
C
C     POSSIBLE DIVERGENCE.  RESIDUAL HAS REMAINED LARGER THAN 10 TIMES
C     THE SMALLEST RESIDUAL FOR NCDS ITERATIONS
   26 IFLAG=4
      GO TO 30
C
C     POSSIBLE DIVERGENCE.  LARGEST SOLUTION INCREMENT NORM HAS
C     INCREASED BY A FACTOR OF 10 FOR THE LAST NCD2S ITERATIONS.
   27 IFLAG=5
      GO TO 30
C
C     MAXIMUM NUMBER OF CONSECUTIVE CUT STEP ITERATIONS TAKEN.  POSSIBLE
C     LOCAL MINIMUM ENCOUNTERED.
   28 IFLAG=6
      GO TO 30
C
C     MATRIX IS SINGULAR.  THE FORMULATION OF THE  THEORY  AND/OR  DERIV
C     SUBROUTINES MAY BE INCORRECT
   29 IFLAG=7
C
   30 CONTINUE
      DO 31 I=1,NPAR
      PAR(I)=OLD(I)
   31 CONTINUE
      RETURN
```

```
      C                                                                    F
      C                                                                    F
   32 FORMAT (2I6,1PE13.5,6E18.10/(25X,6E18.10))                           F
   33 FORMAT (46H0ITERATE  OUT    RESIDUAL  P A R A M E T E R S)           F
      END                                                                  F
```

```
      SUBROUTINE INVRT (A,X1)                                          FI
      DIMENSION A(6,6), X1(6)                                          FI
      COMMON /FIT/ X(2000),Y(2000),WF(2000),WORK(2500),THEOR(2000),NDATA FI
     1,RELERR,ABSERR,IFLAG,PAR(6),R,NPAR                               FI
      EQUIVALENCE (R,RES)                                              FI
      IFLAG=1                                                          FI
      DO 4 I=1,NPAR                                                    FI
      DO 1 J=1,NPAR                                                    FI
      X1(J)=A(I,J)                                                     FI
    1 CONTINUE                                                         FI
      X1(I)=1.0                                                        FI
      IF (ABS(A(I,I)).LT.1.E-99) GO TO 5                               FI
      DO 2 J=1,NPAR                                                    FI
      X1(J)=X1(J)/A(I,I)                                               FI
    2 CONTINUE                                                         FI
      DO 3 J=1,NPAR                                                    FI
      TEMP=A(J,I)                                                      FI
      A(J,I)=0.0                                                       FI
      DO 3 K=1,NPAR                                                    FI
      A(J,K)=A(J,K)-TEMP*X1(K)                                         FI
    3 CONTINUE                                                         FI
      DO 4 K=1,NPAR                                                    FI
    4 A(I,K)=X1(K)                                                     FI
      RETURN                                                           FI
    5 IFLAG=7                                                          FI
      RETURN                                                           FI
      END                                                              FI
```

VARIABLES USED IN LAUNCH

ABSERR - Absolute error tolerance for parameter convergence

ALN - Number of data entries loaded in the vehicle array after processing by VEHICLE

ARAT - Percentage of failures occurring during ascent

BCONT - Name of booster vehicle contractor

BOOST - Booster designator

BV, BV1, BV2 - Booster designators for requesting output information

BVEH - Name of booster vehicle

C1, C2, C3, C4 - Designators for curve fit equations

DERIV1, DERIV2, DERIV3, DERIV4 - Subroutines defining first and second partial derivatives of the fit expression with respect to the parameters at each of the data points.

DERIV - Dummy name in FITIT and FITI to call desired partial derivative subroutine

FA - Ascent failure counter

FAIL - Total failure counter

FL - Land failure counter

FO - Orbital failure counter

FP - Pad failure counter

HEAD - Page heading array

ID1, ID2, ID3 - Launch date separated into month, day, year for printing output

IDATE - Launch date

IFLAG - Duplicate file indicator in RENMER; Error indication flag in CURVIT

IN - File designator

IO - File designator

KEY - Output request parameter indicating desired output subroutines

LATYP - Launch type nomenclature

LOC - Phase of failure

LRAT - Percentage of failures occurring over land

LTYP - Launch type designator

MULTLA - Multiple launch indicator

NDATA - Number of data points in the X and Y arrays to be fit

NEWDAT - Number of data entry cards read

NO - Percentage of launches to be considered in output subroutines

NO1 - Number of launches to be considered in output subroutines

NPAR - Number of parameters in the PAR array

ORAT - Percentage of failures occurring during orbital phase

ORGDAT - Number of data entries in updated main data file

PAR - Parameter values

PF - First partial derivative array

PPF - Second partial derivative array

PRAT - Percentage of failures occurring on the launch pad

PRODIR - Project director name

R - Residual value

RAT - Success ratio

RELERR - Relative error tolerance for parameter convergence

RES - Launch result

RESU - Launch result designator for use in output subroutines

SCR - Dummy variable used for switching data entries

SOUR - Source of information designator as indicated on data entry cards

ST - Source of information designator as stored in main data file

STG - Failed stage designator

SUCC - Success Counter

THEOR1, THEOR2, THEOR3, THEOR4 - Subroutines defining the theoretical fit values at given X values

THEORY - Dummy name in FITIT and FITI to call desired fit subroutine

UCONT - Name of upper stage contractor

UPPER - Upper stage designator

UV, UV1, UV2 - Upper stage designators for requesting output information

UVEH - Name of upper stage

WF - Weighting factors for curve fit data

WORK - Dummy work array

X - Abscissa data variables (launch number)

Y - Ordinate data variables (historical reliability)

All other variables are dummy variables.

ABBREVIATIONS

AFETR	-	Air Force Eastern Test Range
AFWL	-	Air Force Weapons Laboratory
ANSI	-	American National Standards Institute, Inc.
ARPA	-	Advanced Research Projects Agency
DOD	-	Department of Defense
EOF	-	End-of-File
ERDA	-	Energy Research and Development Administration
INSRP	-	Interagency Nuclear Safety Review Panel
LES 8/9	-	Lincoln Experimental Satellites 8 and 9
MJS	-	Mariner Jupiter/Saturn
NASA	-	National Aeronautics and Space Administration
SER	-	Safety Evaluation Report
VAFB	-	Vandenberg Air Force Base

REFERENCES

1. Head, J. H., FITIT, A Computer Program to Least Squares Fit Non-Linear Theories, AFA-TR-70-5, United States Air Force Academy, CO, December 1970.

2. Schefter, J., etc., TRW Space Log, TRW Systems Group, Redondo Beach, CA, various editions.

3. NASA Pocket Statistics, National Aeronautics and Space Administration, Washington, DC, July 1973.

4. Vandenberg AFB Launch Summary, 1st Strategic Aerospace Division, Vandenberg Air Force Base, CA, 1 July 1975.

5. Yanke, A. B., Atlas Flight Program, Summary, General Dynamics-Convair Aerospace Division, San Diego, CA, 21 July 1974.

6. Ames, W. W., Thor The Workhorse of Space A Narrative History, MDC G3770, McDonnell Douglas Astronautics Company, Huntington Beach, CA, 31 July 1972.